

1 INTERNET-DRAFT  
2 draft-ietf-ipp-model-10.txt

R. deBry  
IBM Corporation  
T. Hastings  
Xerox Corporation  
R. Herriot  
Sun Microsystems  
S. Isaacson  
Novell, Inc.  
P. Powell  
San Diego State University  
June 19, 1998

3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15 Internet Printing Protocol/1.0: Model and Semantics  
16 Copyright (C) The Internet Society (date). All Rights Reserved.

17 Status of this Memo

18 This document is an Internet-Draft. Internet-Drafts are working documents of the Internet Engineering  
19 Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute  
20 working documents as Internet-Drafts.

21 Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or  
22 obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material  
23 or to cite them other than as "work in progress".

24 To learn the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in  
25 the Internet-Drafts Shadow Directories on ftp.is.co.za (Africa), nic.nordu.net (Europe), munnari.oz.au  
26 (Pacific Rim), ds.internic.net (US East Coast), or ftp.isi.edu (US West Coast).

27 Abstract

28 This document is one of a set of documents, which together describe all aspects of a new Internet  
29 Printing Protocol (IPP). IPP is an application level protocol that can be used for distributed printing  
30 using Internet tools and technologies. The protocol is heavily influenced by the printing model  
31 introduced in the Document Printing Application (DPA) [ISO10175] standard. Although DPA specifies  
32 both end user and administrative features, IPP version 1.0 (IPP/1.0) focuses only on end user  
33 functionality.

34 The full set of IPP documents includes:

- 35 Requirements for an Internet Printing Protocol [IPP-REQ]
- 36 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]
- 37 Internet Printing Protocol/1.0: Model and Semantics (this document)
- 38 Internet Printing Protocol/1.0: ~~Protocol Specification~~Transport and Encoding [IPP-PRO]

39  
40 The requirements document, "Requirements for an Internet Printing Protocol", takes a broad look at  
41 distributed printing functionality, and it enumerates real-life scenarios that help to clarify the features that  
42 need to be included in a printing protocol for the Internet. It identifies requirements for three types of  
43 users: end users, operators, and administrators. The requirements document calls out a subset of end  
44 user requirements that ~~MUST-beare~~ satisfied in IPP/1.0. Operator and administrator requirements are  
45 out of scope for version 1.0. The rationale document, "Rationale for the Structure and Model and  
46 Protocol for the Internet Printing Protocol", describes IPP from a high level view, defines a roadmap for  
47 the various documents that form the suite of IPP specifications, and gives background and rationale for  
48 the IETF working group's major decisions. This document, "Internet Printing Protocol/1.0: Model and  
49 Semantics", describes a simplified model with abstract objects, their attributes, and their operations. The  
50 model introduces a Printer and a Job. The Job supports multiple documents per Job. The model  
51 document also addresses how security, internationalization, and directory issues are addressed. The  
52 protocol specification, " Internet Printing Protocol/1.0: ~~Protocol Specification~~Transport and Encoding",  
53 is a formal mapping of the abstract operations and attributes defined in the model document onto  
54 HTTP/1.1. The protocol specification defines the encoding rules for a new Internet media type called  
55 "application/ipp".

## 56 Table of Contents

57	1.	Introduction.....	9
58	1.1	Simplified Printing Model.....	11
59	2.	IPP Objects .....	13
60	2.1	Printer Object.....	13
61	2.2	Job Object.....	15
62	2.3	Object Relationships.....	16
63	2.4	Object Identity .....	17
64	3.	IPP Operations .....	20
65	3.1	Common Semantics .....	21
66	3.1.1	Required Elements.....	21
67	3.1.2	Operation IDs and Request IDs .....	21
68	3.1.3	Attributes .....	22
69	3.1.4	Character Set and Natural Language Operation Attributes .....	23
70	3.1.4.1	Request Operation Attributes.....	24
71	3.1.4.2	Response Operation Attributes .....	26
72	3.1.5	Operation Targets.....	27
73	3.1.6	Operation Status Codes and Messages .....	29
74	3.1.7	Versions .....	30
75	3.1.8	Job Creation Operations .....	31
76	3.2	Printer Operations.....	33
77	3.2.1	Print-Job Operation .....	33
78	3.2.1.1	Print-Job Request .....	33
79	3.2.1.2	Print-Job Response .....	37
80	3.2.2	Print-URI Operation .....	40
81	3.2.3	Validate-Job Operation.....	40
82	3.2.4	Create-Job Operation.....	41
83	3.2.5	Get-Printer-Attributes Operation .....	41
84	3.2.5.1	Get-Printer-Attributes Request .....	41
85	3.2.5.2	Get-Printer-Attributes Response .....	43
86	3.2.6	Get-Jobs Operation.....	44
87	3.2.6.1	Get-Jobs Request.....	44
88	3.2.6.2	Get-Jobs Response .....	46
89	3.3	Job Operations.....	47
90	3.3.1	Send-Document Operation.....	48
91	3.3.1.1	Send-Document Request .....	48
92	3.3.1.2	Send-Document Response .....	50
93	3.3.2	Send-URI Operation.....	51
94	3.3.3	Cancel-Job Operation .....	51

95	3.3.3.1	Cancel-Job Request .....	51
96	3.3.3.2	Cancel-Job Response .....	52
97	3.3.4	Get-Job-Attributes Operation.....	52
98	3.3.4.1	Get-Job-Attributes Request .....	53
99	3.3.4.2	Get-Job-Attributes Response .....	54
100	4.	Object Attributes .....	55
101	4.1	Attribute Syntaxes.....	55
102	4.1.1	'text'.....	56
103	4.1.1.1	'textWithoutLanguage'.....	57
104	4.1.1.2	'textWithLanguage'.....	57
105	4.1.2	'name'.....	58
106		'nameWithoutLanguage'.....	59
107	4.1.2.2	'nameWithLanguage'.....	59
108	4.1.3	'keyword'.....	60
109	4.1.4	'enum'.....	61
110	4.1.5	'uri'.....	61
111	4.1.6	'uriScheme'.....	62
112	4.1.7	'charset'.....	62
113	4.1.8	'naturalLanguage'.....	63
114	4.1.9	'mimeMediaType'.....	64
115	4.1.10	'octetString'.....	65
116	4.1.11	'boolean'.....	65
117	4.1.12	'integer'.....	65
118	4.1.13	'rangeOfInteger'.....	65
119	4.1.14	'dateTime'.....	66
120	4.1.15	'resolution'.....	66
121	4.1.16	'1setOf X'.....	66
122	4.2	Job Template Attributes .....	66
123	4.2.1	job-priority (integer(1:100)).....	70
124	4.2.2	job-hold-until (type3 keyword   name (MAX)).....	71
125	4.2.3	job-sheets (type3 keyword   name(MAX)).....	72
126	4.2.4	multiple-document-handling (type2 keyword).....	72
127	4.2.5	copies (integer(1:MAX)).....	73
128	4.2.6	finishings (1setOf type2 enum).....	73
129	4.2.7	page-ranges (1setOf rangeOfInteger (1:MAX)).....	74
130	4.2.8	sides (type2 keyword).....	75
131	4.2.9	number-up (integer(1:MAX)).....	76
132	4.2.10	orientation-requested (type2 enum).....	76
133	4.2.11	media (type3 keyword   name(MAX)).....	77
134	4.2.12	printer-resolution (resolution).....	78

135	4.2.13	print-quality (type2 enum).....	78
136	4.3	Job Description Attributes.....	78
137	4.3.1	job-uri (uri).....	80
138	4.3.2	job-id (integer(1:MAX)) .....	80
139	4.3.3	job-printer-uri (uri) .....	80
140	4.3.4	job-more-info (uri).....	81
141	4.3.5	job-name (name(MAX)) .....	81
142	4.3.6	job-originating-user-name (name(MAX)).....	81
143	4.3.7	job-state (type1 enum) .....	81
144	4.3.8	job-state-reasons (1setOf type2 keyword) .....	84
145	4.3.9	job-state-message (text(MAX)) .....	86
146	4.3.10	number-of-documents (integer(0:MAX)) .....	87
147	4.3.11	output-device-assigned (name(127)) .....	87
148	4.3.12	time-at-creation (integer(0:MAX)).....	87
149	4.3.13	time-at-processing (integer(0:MAX)).....	87
150	4.3.14	time-at-completed (integer(0:MAX)) .....	87
151	4.3.15	number-of-intervening-jobs (integer(0:MAX)) .....	87
152	4.3.16	job-message-from-operator (text(127)).....	88
153	4.3.17	job-k-octets (integer(0:MAX)).....	88
154	4.3.18	job-impressions (integer(0:MAX)) .....	88
155	4.3.19	job-media-sheets (integer(0:MAX)) .....	89
156	4.3.20	job-k-octets-processed (integer(0:MAX)) .....	89
157	4.3.21	job-impressions-completed (integer(0:MAX)).....	89
158	4.3.22	job-media-sheets-completed (integer(0:MAX)) .....	90
159	4.3.23	attributes-charset (charset).....	90
160	4.3.24	attributes-natural-language (naturalLanguage) .....	90
161	4.4	Printer Description Attributes.....	90
162	4.4.1	printer-uri-supported (1setOf uri).....	92
163	4.4.2	uri-security-supported (1setOf type2 keyword) .....	93
164	4.4.3	printer-name (name(127)) .....	94
165	4.4.4	printer-location (text(127)) .....	94
166	4.4.5	printer-info (text(127)) .....	94
167	4.4.6	printer-more-info (uri) .....	94
168	4.4.7	printer-driver-installer (uri) .....	95
169	4.4.8	printer-make-and-model (text(127)).....	95
170	4.4.9	printer-more-info-manufacturer (uri).....	95
171	4.4.10	printer-state (type1 enum).....	95
172	4.4.11	printer-state-reasons (1setOf type2 keyword).....	96
173	4.4.12	printer-state-message (text(MAX)) .....	98
174	4.4.13	operations-supported (1setOf type2 enum).....	99

175	4.4.14	charset-configured (charset).....	99
176	4.4.15	charset-supported (1setOf charset).....	100
177	4.4.16	natural-language-configured (naturalLanguage) .....	100
178	4.4.17	generated-natural-language-supported (1setOf naturalLanguage) .....	100
179	4.4.18	document-format-default (mimeType) .....	101
180	4.4.19	document-format-supported (1setOf mimeType) .....	101
181	4.4.20	printer-is-accepting-jobs (boolean).....	101
182	4.4.21	queued-job-count (integer(0:MAX)).....	101
183	4.4.22	printer-message-from-operator (text(127)).....	101
184	4.4.23	color-supported (boolean).....	102
185	4.4.24	reference-uri-schemes-supported (1setOf uriScheme).....	102
186	4.4.25	pdl-override-supported (type2 keyword).....	102
187	4.4.26	printer-up-time (integer(1:MAX)).....	103
188	4.4.27	printer-current-time (dateTime) .....	103
189	4.4.28	multiple-operation-time-out (integer(1:MAX)).....	103
190	4.4.29	compression-supported (1setOf type3 keyword) .....	103
191	4.4.30	job-k-octets-supported (rangeOfInteger(0:MAX)) .....	104
192	4.4.31	job-impressions-supported (rangeOfInteger(0:MAX)).....	104
193	4.4.32	job-media-sheets-supported (rangeOfInteger(0:MAX)).....	104
194	5.	Conformance .....	104
195	5.1	Client Conformance Requirements .....	104
196	5.2	IPP Object Conformance Requirements.....	105
197	5.2.1	Objects .....	105
198	5.2.2	Operations .....	105
199	5.2.3	IPP Object Attributes.....	106
200	5.2.4	Extensions .....	106
201	5.2.5	Attribute Syntaxes .....	107
202	5.3	Charset and Natural Language Requirements .....	107
203	5.4	Security Conformance Requirements.....	107
204	6.	IANA Considerations (registered and private extensions).....	107
205	6.1	Typed 'keyword' and 'enum' Extensions .....	108
206	6.2	Attribute Extensibility .....	110
207	6.3	Attribute Syntax Extensibility .....	111
208	6.4	Operation Extensibility .....	111
209	6.5	Status Code Extensibility .....	111
210	6.6	Registration of MIME types/sub-types for document-formats.....	112
211	6.7	Registration of charsets for use in 'charset' attribute values .....	112
212	7.	Internationalization Considerations .....	112
213	8.	Security Considerations .....	115
214	8.1	Security Scenarios.....	116

215	8.1.1	Client and Server in the Same Security Domain.....	116
216	8.1.2	Client and Server in Different Security Domains.....	116
217	8.1.3	Print by Reference .....	117
218	8.2	URIs for TLS and non-TLS Access.....	117
219	8.3	The "requesting-user-name" (name(MAX)) Operation Attribute.....	117
220	8.4	Restricted Queries.....	119
221	8.5	IPP Security Application Profile for TLS.....	119
222	9.	References.....	120
223	10.	Copyright Notice.....	123
224	11.	Author's Address.....	123
225	12.	Formats for IPP Registration Proposals .....	127
226	12.1	Type2 keyword attribute values registration .....	127
227	12.2	Type3 keyword attribute values registration .....	127
228	12.3	Type2 enum attribute values registration .....	128
229	12.4	Type3 enum attribute values registration .....	128
230	12.5	Attribute registration.....	128
231	12.6	Attribute Syntax registration .....	129
232	12.7	Operation registration .....	129
233	12.8	Status code registration.....	129
234	13.	APPENDIX A: Terminology .....	130
235	13.1	Conformance Terminology.....	130
236	13.1.1	MUST .....	130
237	13.1.2	MUST NOT .....	130
238	13.1.3	SHOULD .....	130
239	13.1.4	SHOULD NOT .....	131
240	13.1.5	MAY .....	131
241	13.1.6	NEED NOT .....	131
242	13.2	Model Terminology .....	131
243	13.2.1	Keyword.....	131
244	13.2.2	Attributes .....	131
245	13.2.2.1	Attribute Name.....	132
246	13.2.2.2	Attribute Group Name.....	132
247	13.2.2.3	Attribute Value.....	132
248	13.2.2.4	Attribute Syntax .....	132
249	13.2.3	Supports.....	132
250	13.2.4	print-stream page.....	134
251	13.2.5	impression .....	134
252	14.	APPENDIX B: Status Codes and Suggested Status Code Messages .....	134
253	14.1	Status Codes.....	135
254	14.1.1	Informational .....	136

255	14.1.2	Successful Status Codes .....	136
256	14.1.2.1	successful-ok (0x0000) .....	136
257	14.1.2.2	successful-ok-ignored-or-substituted-attributes (0x0001) .....	136
258	14.1.2.3	successful-ok-conflicting-attributes (0x0002) .....	136
259	14.1.3	Redirection Status Codes .....	136
260	14.1.4	Client Error Status Codes .....	137
261	14.1.4.1	client-error-bad-request (0x0400) .....	137
262	14.1.4.2	client-error-forbidden (0x0401) .....	137
263	14.1.4.3	client-error-not-authenticated (0x0402) .....	137
264	14.1.4.4	client-error-not-authorized (0x0403) .....	137
265	14.1.4.5	client-error-not-possible (0x0404) .....	138
266	14.1.4.6	client-error-timeout (0x0405) .....	138
267	14.1.4.7	client-error-not-found (0x0406) .....	138
268	14.1.4.8	client-error-gone (0x0407) .....	138
269	14.1.4.9	client-error-request-entity-too-large (0x0408) .....	139
270	14.1.4.10	client-error-request-value-too-long (0x0409) .....	139
271	14.1.4.11	client-error-document-format-not-supported (0x040A) .....	139
272	14.1.4.12	client-error-attributes-or-values-not-supported (0x040B) .....	139
273	14.1.4.13	client-error-uri-scheme-not-supported (0x040C) .....	140
274	14.1.4.14	client-error-charset-not-supported (0x040D) .....	140
275	14.1.4.15	client-error-conflicting-attributes (0x040E) .....	140
276	14.1.5	Server Error Status Codes .....	140
277	14.1.5.1	server-error-internal-error (0x0500) .....	140
278	14.1.5.2	server-error-operation-not-supported (0x0501) .....	140
279	14.1.5.3	server-error-service-unavailable (0x0502) .....	141
280	14.1.5.4	server-error-version-not-supported (0x0503) .....	141
281	14.1.5.5	server-error-device-error (0x0504) .....	141
282	14.1.5.6	server-error-temporary-error (0x0505) .....	141
283	14.1.5.7	server-error-not-accepting-jobs (0x0506) .....	142
284	14.1.5.8	server-error-busy (0x0507) .....	142
285	14.2	Status Codes for IPP Operations .....	143
286	15.	APPENDIX C: "media" keyword values .....	143
287	16.	APPENDIX D: Processing IPP Attributes .....	148
288	16.1	Fidelity .....	148
289	16.2	Page Description Language (PDL) Override .....	149
290	16.3	Suggested Operation Processing Steps for All Operations .....	151
291	16.3.1	Validate version number .....	151
292	16.3.2	Validate operation identifier .....	152
293	16.3.3	Validate the request identifier .....	152
294	16.3.4	Validate attribute group and attribute presence and order .....	152



295	16.3.4.1	Validate the presence and order of attribute groups.....	152
296	16.3.4.2	Ignore unknown attribute groups in the expected position.....	152
297	16.3.4.3	Validate the presence of a single occurrence of required Operation attributes .....	153
298	16.3.5	Validate the values of the MANDATORY Operation attributes .....	159
299	16.3.6	Validate the values of the OPTIONAL Operation attributes .....	162
300	16.4	Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add Documents.....	164
301			
302	16.4.1	Default "ipp-attribute-fidelity" if not supplied.....	164
303	16.4.2	Check that the Printer object is accepting jobs.....	164
304	16.4.3	Validate the values of the Job Template attributes.....	164
305	16.4.4	Check for conflicting Job Template attributes values .....	168
306	16.4.5	Decide whether to REJECT the request.....	169
307	16.4.6	For the Validate-Job operation, RETURN one of the success status codes.....	169
308	16.4.7	Create the Job object with attributes to support .....	170
309	16.4.8	Return one of the success status codes.....	171
310	16.4.9	Accept appended Document Content .....	171
311	16.4.10	Scheduling and Starting to Process the Job .....	171
312	16.4.11	Completing the Job.....	172
313	16.4.12	Destroying the Job after completion.....	172
314	16.4.13	Interaction with "ipp-attribute-fidelity" .....	172
315	16.5	Using Job Template Attributes During Document Processing. ....	172
316	17.	APPENDIX E: Generic Directory Schema.....	174

## 317 1. Introduction

318 The Internet Printing Protocol (IPP) is an application level protocol that can be used for distributed  
 319 printing using Internet tools and technologies. IPP version 1.0 (IPP/1.0) focuses only on end user  
 320 functionality. This document is just one of a suite of documents that fully define IPP. The full set of IPP  
 321 documents includes:

- 322 Requirements for an Internet Printing Protocol [IPP-REQ]
- 323 Rationale for the Structure and Model and Protocol for the Internet Printing Protocol [IPP-RAT]
- 324 Internet Printing Protocol/1.0: Model and Semantics (this document)
- 325 Internet Printing Protocol/1.0: ~~Protocol Specification~~[Transport and Encoding](#) [IPP-PRO]

326  
 327 Anyone reading this document for the first time is strongly encouraged to read the IPP documents in the  
 328 following order:

- 329 1. The requirements document, "Requirements for an Internet Printing Protocol". That document  
 330 takes a broad look at distributed printing functionality, and it enumerates real-life scenarios that

- 331 help to clarify the features that need to be included in a printing protocol for the Internet. It  
332 identifies requirements for three types of users: end users, operators, and administrators. The  
333 requirements document calls out a subset of end user requirements that ~~MUST be~~ are satisfied in  
334 IPP/1.0. Operator and administrator requirements are out of scope for version 1.0.
- 335 2. The rationale document, "Rationale for the Structure and Model and Protocol for the Internet  
336 Printing Protocol". That document describes IPP from a high level view, defines a roadmap for  
337 the various documents that form the suite of IPP specifications, and gives background and  
338 rationale for the IETF working group's major decisions.
- 339 3. This document, the "Internet Printing Protocol/1.0: Model and Semantics" document. ~~It~~ This  
340 document describes a simplified model with abstract objects, their attributes, and their operations.  
341 The model introduces a Printer and a Job. ~~The~~ A Job optionally supports multiple documents per  
342 Job. The model document also ~~addresses~~ describes how security, internationalization, and  
343 directory issues are addressed.
- 344 4. The protocol specification, " Internet Printing Protocol/1.0: ~~Protocol Specification~~ Transport and  
345 Encoding". That document is a formal mapping of the abstract operations and attributes defined  
346 in the model document onto HTTP/1.1. The protocol specification defines the encoding rules for  
347 a new Internet media type called "application/ipp".

348  
349 This document is laid out as follows:

- 350 - The rest of Section 1 is an introduction to the IPP simplified model for distributed printing.  
351 - Section 2 introduces the object types covered in the model ~~and~~ with their basic behaviors, attributes,  
352 and interactions.
- 353 - Section 3 defines the operations ~~supported by~~ included in IPP/1.0. IPP operations are synchronous,  
354 therefore, for each operation, there is a both request and a response.
- 355 - Section 4 defines the attributes (and their syntaxes) that are used in the model.
- 356 - Sections 5 - 6 summarizes the implementation conformance requirements for objects that support  
357 the protocol and IANA considerations, respectively.
- 358 - Sections 7 - 11 cover the Internationalization and Security considerations as well as References,  
359 Copyright Notice, and Author contact information.
- 360 - Sections 12 - 14 are appendices that cover Terminology, Status Codes and Messages, and "media"  
361 keyword values. This document uses terms such as "attributes", "keywords", and "support".  
362 These terms have special meaning and are defined in the model terminology section. Capitalized  
363 terms such as MANDATORY, SHALL, and OPTIONAL have special meaning relating to  
364 conformance. These terms are defined in the section on conformance terminology, most of which  
365 is taken from RFC 2119 [RFC2119].
- 366 - Section 15 is an appendix that defines the rules and suggested techniques for the processing of  
367 attributes in client requests by IPP objects. This section helps to clarify the ~~ae~~ effects of interactions  
368 between related attributes and their values.

369 - Section 16 is an appendix that enumerates athe subset of Printer attributes that form a generic  
370 directory schema. These attributes are useful when registering a Printer so that a client can find  
371 the Printer not just by name, but by filtered searches as well.

## 372 1.1 Simplified Printing Model

373 In order to achieve its goal of realizing a workable printing protocol for the Internet, the Internet Printing  
374 Protocol (IPP) is based on a simplified printing model that abstracts the many components of real world  
375 printing solutions. The Internet is a distributed computing environment where requesters of print services  
376 (clients, applications, printer drivers, etc.) cooperate and interact with print service providers. This model  
377 and semantics document describes a simple, abstract model for IPP even though the underlying  
378 configurations may be complex "n-tier" client/server systems. An important simplifying step in the IPP  
379 model is to expose only the key objects and interfaces required for printing. The model described in this  
380 model document does not include features, interfaces, and relationships that are beyond the scope of the  
381 first version of IPP (IPP/1.0). IPP/1.0 incorporates many of the relevant ideas and lessons learned from  
382 other specification and development efforts [HTPP] [ISO10175] [LDPA] [P1387.4] [PSIS] [RFC1179]  
383 [SWP].

384 The IPP/1.0 model encapsulates the important components of distributed printing into two object types:

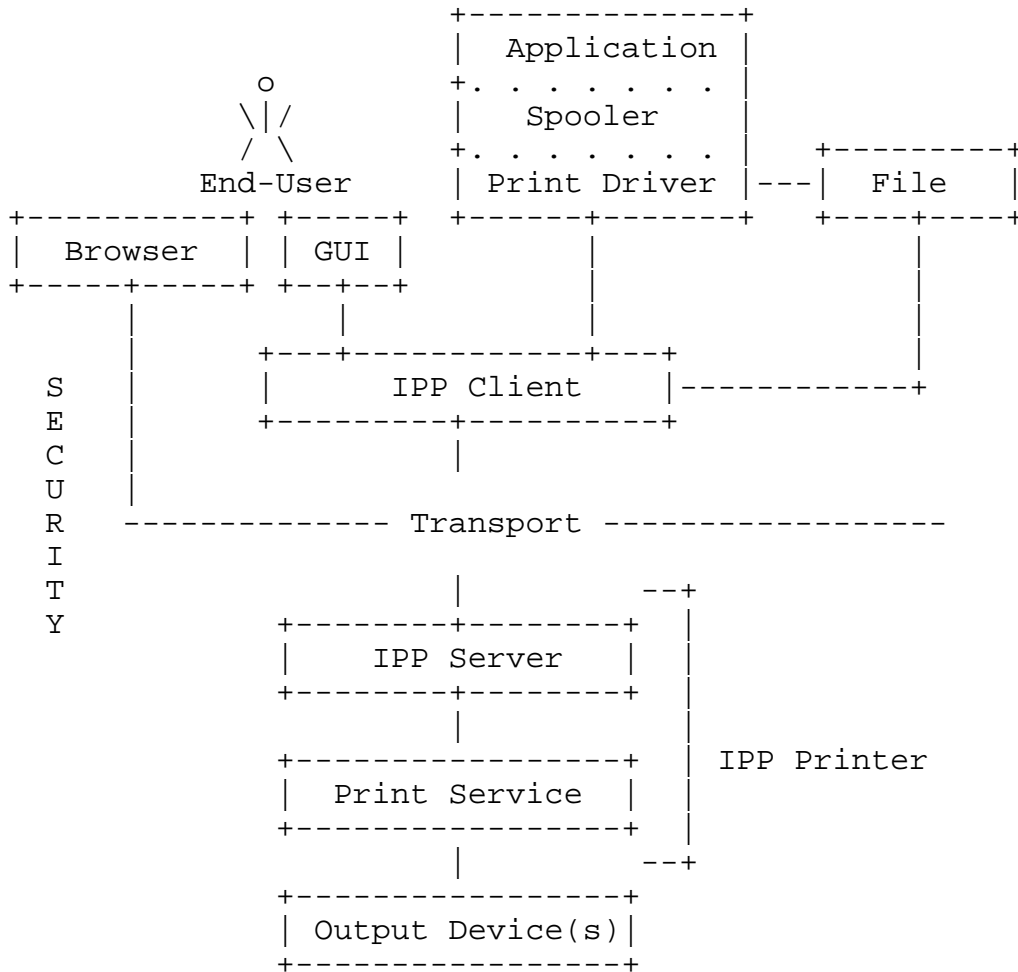
- 385 - Printer (Section 2.1)
- 386 - Job (Section 2.2)

387  
388 Each object type has an associated set of operations (see section 3) and attributes (see section 4).

389 It is important, however, to understand that in real system implementations (which lie underneath the  
390 abstracted IPP/1.0 model), there are other components of a print service which are not explicitly defined  
391 in the IPP/1.0 model. The following figure illustrates where IPP/1.0 fits with respect to these other  
392 components.

393  
394  
395  
396  
397  
398  
399  
400  
401  
402  
403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423

N D S  
O I E  
T R C  
I E U  
F C R  
I T I  
C O T  
A R Y  
T Y  
I  
O  
N



An IPP Printer object encapsulates the functions normally associated with physical output devices along with the spooling, scheduling and multiple device management functions often associated with a print server. Printer objects are optionally registered as entries in a directory where end users find and select them based on some sort of filtered and context based searching mechanism (see section 17). The directory is used to store relatively static information about the Printer, allowing end users to search for and find Printers that match their search criteria, for example: name, context, printer capabilities, etc.. The more dynamic information, such as state, currently loaded and ready media, number of jobs at the Printer, errors, warnings, and so forth, is directly associated with the Printer object itself rather than with the ~~(as compared to the~~ entry in the directory which only represents the Printer object). ~~This more dynamic information includes state, currently loaded and ready media, number of jobs at the Printer, errors, warnings, and so forth.~~

435 IPP clients implement the IPP protocol on the client side, and give end users (or programs running on  
436 behalf of end users) the ability to query Printer objects and submit and manage print jobs. An IPP server  
437 is just that part of the Printer object that implements the server-side protocol. The rest of the Printer  
438 object implements (or gateways into) the application semantics of the print service itself. The Printer  
439 objects may be embedded in an output device or may be implemented on a host on the network that  
440 communicates with ~~the-an~~ output device.

441 When a job is submitted to the Printer object and the Printer object validates the attributes in the  
442 submission request, the Printer object creates a new Job object. The end user then interacts with this new  
443 Job object to query its status and monitor the progress of the job. End users may also cancel the print job  
444 by using the Job object's Cancel-Job operation. The notification service ~~(s) are-is~~ out of scope for  
445 IPP/1.0, but using such a notification service, the end user is able to register for and receive Printer  
446 specific and Job specific events. An end user can query the status of Printer objects and can follow the  
447 progress of Job objects by polling using the Get-Printer-Attributes, Get-Jobs, and Get-Job-Attributes  
448 operations.

## 449 2. IPP Objects

450 The IPP/1.0 model introduces objects of type Printer and Job. Each type of object models relevant  
451 aspects of a real-world entity such as a real printer or real print job. Each object type is defined as a set  
452 of possible attributes that may be supported by instances of that object type. For each object (instance),  
453 the actual set of supported attributes and values describe a specific implementation. The object's  
454 attributes and values describe its state, capabilities, realizable features, job processing functions, and  
455 default behaviors and characteristics. For example, the Printer object type is defined as a set of attributes  
456 that each Printer object potentially supports. In the same manner, the Job object type is defined as a set  
457 of attributes that are potentially supported by each Job object.

458 Each attribute included in the set of attributes defining an object type is labeled as:

- 459 - "MANDATORY": each object SHALL support the attribute.
- 460 - "OPTIONAL": each object ~~OPTIONALLY-MAY~~ supports the attribute.

461

462 There is no such similar labeling of attribute values. However, if an implementation supports an attribute,  
463 it MUST support at least one of the possible values for that attribute.

### 464 2.1 Printer Object

465 ~~A-The~~ major component of the IPP/1.0 model is the Printer object. A Printer object implements the  
466 server-side of the IPP/1.0 protocol. Using the protocol, end users may query the attributes of the Printer

467 object and submit print jobs to the Printer object. The actual implementation components behind the  
468 Printer abstraction may take on different forms and different configurations. However, the model  
469 abstraction allows the details of the configuration of real components to remain opaque to the end user.  
470 Section 3 describes each of the Printer operations in detail.

471 The capabilities and state of a Printer object are described by its attributes. Printer attributes are divided  
472 into two groups:

- 473 - "job-template" attributes: These attributes describe supported job processing capabilities and  
474 defaults for the Printer object. (See section 4.2)
- 475 - "printer-description" attributes: These attributes describe the Printer object's identification, state,  
476 location, references to other sources of information about the Printer object, etc. (see section 4.4)  
477

478 Since a Printer object is an abstraction of a generic document output device and print service provider, a  
479 Printer object could be used to represent any real or virtual device with semantics consistent with the  
480 Printer object, such as a fax device, an imager, or even a CD writer.

481 Some examples of configurations supporting a Printer object include:

- 482 1) An output device, with no spooling capabilities
- 483 2) An output device, with a built-in spooler
- 484 3) A print server supporting IPP with one or more associated output devices
  - 485 3a) The associated output devices ~~might or might~~may or may not be capable of spooling jobs
  - 486 3b) The associated output devices ~~might or might~~may or may not support IPP

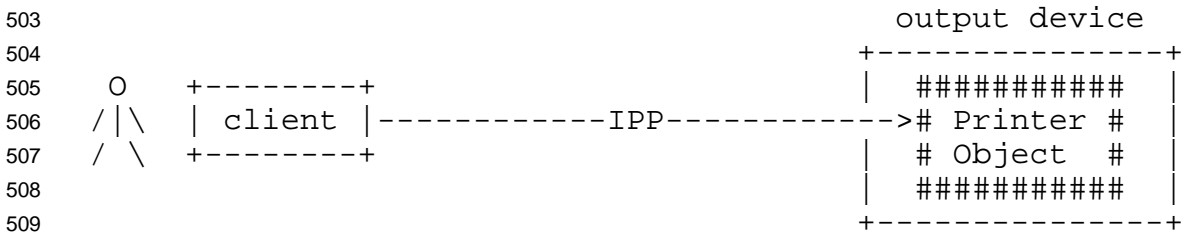
487  
488 The following figures show some examples of how Printer objects can be realized on top of various  
489 distributed printing configurations. The embedded case below represents configurations 1 and 2. The  
490 hosted and fan-out figures below represent configurations 3a and 3b.

491 Legend:

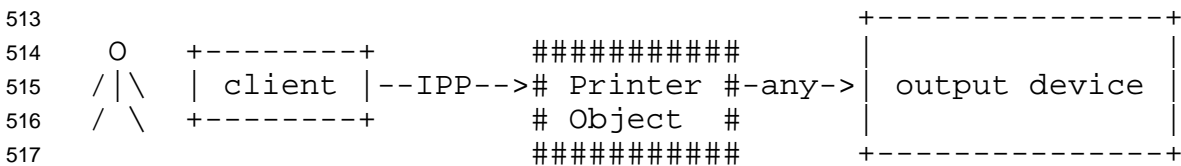
492  
 493 ##### indicates a Printer object which is  
 494 either embedded in an output device or is  
 495 hosted in a server. The Printer object  
 496 might or might not be capable of queuing/spooling.  
 497

498 any indicates any network protocol or direct  
 499 connect, including IPP  
 500

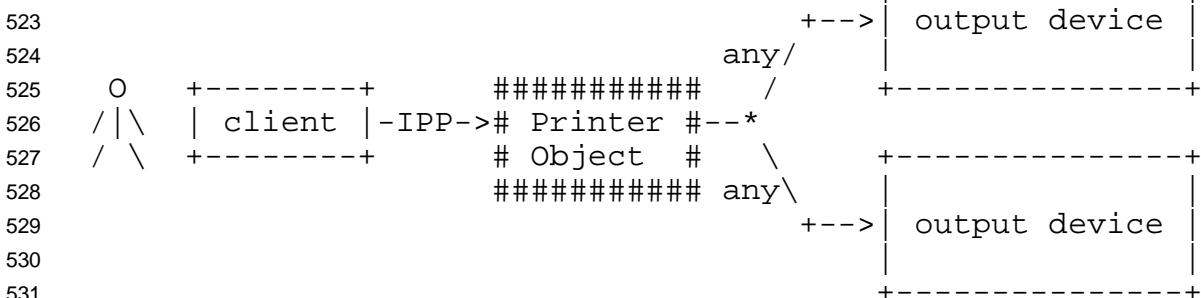
501 embedded printer:



511 hosted printer:



521 fan out:



534 2.2 Job Object

535 A Job object is used to model a print job. A Job can contain one or more documents. The information  
536 required to create a Job object is sent in a create request from the end user via an IPP Client to the Printer  
537 object. The Printer object validates the create request, and if the Printer object accepts the request, the  
538 Printer object creates the new Job object. Section 3 describes each of the Job operations in detail.

539 The characteristics and state of a Job object are described by its attributes. Job attributes are grouped  
540 into two groups as follows:

- 541 - "job-template" attributes: These attributes are ~~OPTIONALLY~~can be supplied by the client or end  
542 user and include job processing instructions which are intended to override any Printer object  
543 defaults and/or instructions embedded within the document data. (See section 4.2)
- 544 - "job-description" attributes: These attributes describe the Job object's identification, state, size, etc.  
545 The client supplies some of these attributes, and the Printer object generates others. (See section  
546 4.3)

547  
548 A Job object contains at least one document, but may contain multiple documents. A document ~~consists~~  
549 of is either:

- 550 - a stream of document data in a format supported by the Printer object (typically a Page Description  
551 Language - PDL), or
- 552 - a reference to such a stream of document data

553  
554 In IPP/1.0, a document is not modeled as an IPP object, therefore it has no object identifier or associated  
555 attributes. All job processing instructions are modeled as Job object attributes. These attributes are  
556 called Job Template attributes and they apply equally to all documents within a Job object.

## 557 2.3 Object Relationships

558 IPP objects have relationships that ~~MUST be~~are maintained persistently along with the persistent storage  
559 of the object attributes.

560 A Printer object can represent either ~~MAY represent~~ one or more physical output devices or ~~-. A Printer~~  
561 ~~object MAY represent~~ a logical device which "processes" jobs but never actually uses a physical output  
562 device to put marks on paper ~~(. Examples of logical devices include for example~~ a Web page publisher or  
563 ~~an interface gateway~~ into an online document archive or repository) ~~-. A Printer object contains zero or~~  
564 more Job objects.

565 A Job object is contained by exactly one Printer object, however the identical document data associated  
566 with a Job object could be sent to either the same or a different Printer object. In this case, a ~~new~~second



567 Job object would be created which would be almost identical to the ~~existing~~first Job object, however it  
568 would have new (different) Job object identifiers (see section 2.4).

569 A Job object contains one or more documents. If the contained document is a stream of document data,  
570 that stream can be contained in only one document. However, there can be identical copies of the stream  
571 in other documents in the same or different Job objects. If the contained document is just a reference to a  
572 stream of document data, other documents (in the same or different Job object(s)) may contain the same  
573 reference, ~~the same stream~~.

## 574 2.4 Object Identity

575 All Printer and Job objects are identified by ~~an~~ Uniform Resource Identifier (URI) [RFC1630] identifier  
576 so that they can be persistently and unambiguously referenced. ~~The IPP/1.0 model suggests that the~~  
577 ~~identifiers for IPP object could be Uniform Resource Identifiers (URIs) [RFC1630]. For example, the~~  
578 ~~IPP model names the Printer and Job object identifying attributes with names like "printer-uri" and "job-~~  
579 ~~uri". The notion of a URI is a useful concept, however, However,~~ until the notion of URI is more stable  
580 (i.e., defined more completely and deployed more widely), it is expected that the URIs used for IPP  
581 objects will actually be URLs [RFC1738] [RFC1808]. ~~As currently defined, Since~~ every URL is a  
582 ~~specialized specialized form of a~~ URI, ~~therefore,~~ even though the more generic term "URI" is used  
583 throughout the rest of this document, its usage is intended to cover ~~to~~ the more specific notion of "URL"  
584 as well.

585 An administrator~~s~~ configures~~s~~ Printer objects to either support or not support authentication and/or  
586 message privacy using TLS [TLS] (the mechanism for security configuration is outside the scope of  
587 IPP/1.0). In some situations, both types of connections (both authenticated and unauthenticated) can be  
588 established and other security mechanisms via using a single communication channel that has some sort  
589 of negotiation mechanism. In other situations, or multiple communication channels are used, one for  
590 each type of security configuration. Section 8 provides a full description of all security considerations  
591 and configurations. ~~The mechanism for such configuration is outside the scope of IPP/1.0. Section 8~~  
592 ~~provides a full description of all security considerations. However, it must be mentioned here that~~

593 ~~if~~ If a Printer object supports more than one communication channel, some or all of those channels might  
594 support and/or require different security mechanisms. In such cases, an administrator could expose the  
595 simultaneous support for these multiple communication channels as ~~A single URI for the Printer object~~  
596 ~~over which clients and Printer objects negotiate to determine a mutually agreeable security mechanism (if~~  
597 ~~any), or Multiple multiple URIs for a single the~~ Printer object where each URI represents each one of the  
598 communication channels to the Printer object. ~~(each with possibly different security characteristics and~~  
599 ~~configurations).~~

600

601 To support this flexibility, the IPP Printer object type defines a multi-valued identification attribute called  
602 the "printer-uri-supported" attribute. It **MUST** contain at least one URI. It **MAY** contain more than one  
603 URI. That is, every Printer object will have at least one URI which identifies at least one communication  
604 channel to the Printer object, ~~it,~~ but it may have more than one URI where each URI identifies a different  
605 communication channel to the Printer object. The "printer-uri-supported" attribute has a companion  
606 attribute, the "uri-security-supported" attribute, that has the same cardinality as "printer-uri-supported".  
607 The purpose of the "uri-security-supported" attribute is to indicate the security mechanisms (if any) used  
608 for each URI listed in "printer-uri-supported". These two attributes are fully described in sections 4.4.1  
609 and 4.4.2.

610 When a job is submitted to the Printer object via a create request, the client ~~MUST supply~~ supplies only a  
611 single Printer object URI. The client supplied Printer object URI **MUST** be one of the values in the  
612 "printer-uri-supported" Printer attribute.

613 Note: IPP/1.0 does not specify how the client obtains the client supplied URI, but it is  
614 **RECOMMENDED** that a Printer object be registered as an entry in a directory service. End-users and  
615 programs can then interrogate the directory searching for Printers. Section 17 defines a generic schema  
616 for Printer object entries in the directory service and describes how the entry acts as a bridge to the actual  
617 IPP Printer object. The entry in the directory that represents the IPP Printer object includes the possibly  
618 many URIs for that Printer object as values ~~of~~ in one its attributes.

619 When a client submits a create request to the Printer object, the Printer object validates the request and  
620 creates a new Job object. The Printer object assigns the new Job object a URI which is stored in the  
621 "job-uri" Job attribute. This URI is then used by clients as the target for subsequent Job operations. The  
622 Printer object generates a Job URI based on its configured security policy and the URI used by the client  
623 in the create request, ~~(if there is more than one URI for the Printer object).~~

624 For example, consider a Printer object that supports both a communication channel secured by the use of  
625 SSL3-TLS (using a standard URI indicating the use of HTTP over TLS) ~~an "https" schemed URI~~ and  
626 another open communication channel that is not secured with SSL3-TLS (using an simple "http" schemed  
627 URI). If a client were to submit a job using the secure URI, the Printer object would assign the new Job  
628 object a secure URI as well. If a client were to submit a job using the open-channel URI, the Printer  
629 would assign the new Job object an open-channel URI.

630 In addition, the Printer object also ~~fills in~~ populates the Job object's "job-printer-uri" attribute. This is a  
631 reference back to the Printer object that created the Job object. If a client only has access to a Job  
632 object's "job-uri" identifier, the client can query the "job-printer-uri" attribute in order to determine which  
633 Printer object created the Job object. If the Printer object supports more than one URI, the Printer object  
634 picks the one URI supplied by the client when creating the job to build the value for and to populate the  
635 "job-printer-uri" attribute.

636 Allowing Job objects to have URIs allows for flexibility and scalability. For example, in some  
637 implementations, the Printer object might create Jobs that are processed in the same local environment as  
638 the Printer object itself. In this case, the Job URI might just be a composition of the Printer's URI (~~the~~  
639 ~~URI used by the submitting client if the Printer object supports more than one URI~~) and some unique  
640 component for the Job object, such as the unique 32-bit positive integer mentioned later in this paragraph.  
641 In other implementations, the Printer object might be a central clearing-house for validating all Job object  
642 creation requests, ~~and-but~~ the Job object itself might be created in some environment that is remote from  
643 the Printer object. In this case, the Job object's URI may have no physical-location relationship at all to  
644 the Printer object's URI. Again, ~~this-the fact that Job objects have URIs~~ allows for flexibility and  
645 scalability, however, many existing printing systems have local models or interface constraints that force  
646 print jobs to be identified using only a 32-bit positive integer rather than an independent URI. This  
647 numeric Job ID is only unique within the context of the Printer object to which the create request was  
648 originally submitted. Therefore, in order to allow both types of client access to IPP Job objects (either by  
649 Job URI or by numeric Job ID), when the Printer object successfully processes a create request and  
650 creates a new Job object, the Printer object SHALL generate both a Job URI and a Job ID. The Job ID  
651 (stored in the "job-id" attribute) only has meaning in the context of the Printer object to which the create  
652 request was originally submitted. This requirement to support both Job URIs and Job IDs allows all types  
653 of clients to access Printer objects and Job objects no matter the local constraints imposed on the client  
654 implementation.

655 In addition to identifiers, Printer objects and Job objects have names. An object name need not be unique  
656 across all instances of all objects. A Printer object's name is chosen and set by an administrator through  
657 some mechanism outside the scope of IPP/1.0. A Job object's name is optionally chosen and supplied by  
658 the IPP client submitting the job. If the client does not supply a Job object name, the Printer object  
659 generates a name for the new Job object. In all cases, the name only has local meaning; ~~the name is not~~  
660 ~~constrained to be unique.~~

661 To summarize:

- 662 - Each Printer object is identified with one or more URIs. The Printer's "printer-uri-supported"  
663 attribute contains the URI(s).
- 664 - The Printer object's "uri-security-supported" attribute identifies the communication channel security  
665 protocols that may or may not have been configured for the various Printer object URIs (e.g., 'tls'  
666 or 'none').
- 667 - Each Job object is identified with a Job URI. The Job's "job-uri" attribute contains the URI.
- 668 - Each Job object is also identified with Job ID which is a 32-bit, positive integer. The Job's "job-id"  
669 attribute contains the Job ID. The Job ID is only unique within the context of the Printer object  
670 (~~using one of its URIs~~) which created the Job object.
- 671 - Each Job object has a "job-printer-uri" attribute which contains the URI of the Printer object that  
672 was used when to creating create the Job object. This attribute is used to determine the Printer  
673 object that created a Job object when given only the URI for the Job object. This linkage is

674 necessary to determine the languages, charsets, and operations which are supported ~~for operations~~  
675 on that Job ~~—~~(the basis for such support comes from the creating Printer object).

- 676 - Each Printer object has a name (which is not necessarily unique). The administrator chooses and  
677 sets this name through some mechanism outside the scope of IPP/1.0 itself. The Printer object's  
678 "printer-name" attribute contains the name.
- 679 - Each Job object has a name (which is not necessarily unique). The client optionally supplies this  
680 name in the create request. If the client does not supply this name, the Printer object generates a  
681 name for the Job object. The Job object's "job-name" attribute contains the name.

### 682 3. IPP Operations

683 IPP objects support operations. An operation consists of a request and a response. When a client  
684 communicates with an IPP object, the client issues an operation request to the URI for that object.  
685 Operations have attributes that supply information about the operation itself. These attributes are called  
686 operation attributes (as compared to object attributes such as Printer object attributes or Job object  
687 attributes). Each request carries along with it any operation attributes, object attributes, and/or document  
688 data required ~~by the object~~ to perform the operation. Each request requires a response from the object.  
689 Each response indicates success or failure of the operation with a status code. The response contains any  
690 operation attributes, object attributes, and/or status messages generated ~~by~~ during the execution of the  
691 operation request.

692 This section describes the semantics of the IPP operations, both requests and responses, in terms of the  
693 attributes and other data associated with each operation.

694 The IPP/1.0 Printer operations are:

- 695 Print-Job (section 3.2.1)
- 696 Print-URI (section 3.2.2)
- 697 Validate-Job (section 3.2.3)
- 698 Create-Job (section 3.2.4)
- 699 Get-Printer-Attributes (section 3.2.5)
- 700 Get-Jobs (section 3.2.6)

701

702 The Job operations are:

- 703 Send-Document (section 3.3.1)
- 704 Send-URI (section 3.3.2)
- 705 Cancel-Job (section 3.3.3)
- 706 Get-Job-Attributes (section 3.3.4)

707

708 The Send-Document and Send-URI Job operations are used to add a new document to an existing multi-  
709 document Job object created ~~with using~~ the Create-Job operation.

### 710 3.1 Common Semantics

711 ~~The following sections describe the~~ All IPP operations share some common elements and features ~~of all~~  
712 IPP operations. Operations are always used in request/response pairs. These common elements are  
713 defined and described in more detail in the following sections.

#### 714 3.1.1 Required Elements

715 Every operation request contains:

- 716 - a "version-number",
- 717 - an "operation-id",
- 718 - a "request-id", and
- 719 - the attributes that are MANDATORY for that type of request.

720

721 Every operation response contains:

- 722 - a "version-number",
- 723 - a "status-code",
- 724 - the "request-id" that was supplied in the corresponding request, and
- 725 - the attributes that are MANDATORY for that type of response.

726

727 Note: The transport and encoding document [IPP-PRO] defines special rules for the encoding of the  
728 "operation-id", the "version-number", the "status-code", and the "request-id". All other operation  
729 elements represented using the more generic encoding rules for attributes and groups of attributes.

#### 730 ~~3.1.13.1.2~~ Operation IDs and Request IDs

731 Each IPP operation ~~request includes an identifying "operation-id" value. Valid values are type is defined~~  
732 ~~as one of the values of the in~~ the "operations-supported" Printer attribute section (see section 4.4.13).  
733 The client specifies which the operation is being requested by supplying the correct by including an  
734 "operation-id" attribute value. ~~The "operation-id" is passed in every request and each corresponding~~  
735 ~~response.~~

736 In addition, every invocation of an operation is identified by a ~~Each operation request also carries with it~~  
737 a "request-id" attribute value. For each request, the client chooses the "request-id" which is an integer  
738 (possibly unique depending on client requirements) in the range from 1 to 2\*\*31 - 1 (inclusive). This  
739 "request-id" attribute allows clients to manage multiple outstanding requests. ~~For each operation request,~~  
740 ~~the client chooses an integer (possibly unique depending on client requirements) in the range from 1 to~~

741 ~~2\*\*31-1 (inclusive).~~ The receiving IPP object, copies the client supplied "request-id" attribute into the  
742 response so ~~that that~~ the client can match ~~up~~ the ~~correct~~ response with the correct outstanding request.

743 Note: In some cases, the transport protocol underneath IPP might be a connection oriented protocol that  
744 would make it impossible for a client to receive responses in any order other than the order in which the  
745 corresponding requests were sent. In such cases, the "request-id" attribute would not be essential for  
746 correct protocol operation. ~~h~~ However, in other mappings, the operation responses can come back in  
747 any order. In these cases, the "request-id" would be essential.

### 748 3.1.23.1.3 Attributes

749 Operation requests and responses are both composed of groups of attributes and/or document data. The  
750 attributes groups are:

- 751 - Operation Attributes: These attributes are passed in the operation and affect the IPP object's  
752 behavior while processing the operation request and ~~MAY~~may affect other attributes or groups  
753 of attributes. Some operation attributes describe the document data associated with the print job  
754 and are associated with new Job objects, however most operation attributes do not persist beyond  
755 the life of the operation. The description of each operation attribute includes conformance  
756 statements indicating which operation attributes are MANDATORY and which are OPTIONAL  
757 for an IPP object to support and which attributes a client MUST supply in a request and an IPP  
758 object MUST supply in a response.
- 759 - Job Template Attributes: These attributes affect the processing of a job. A client OPTIONALLY  
760 supplies Job Template Attributes in a create request, and the receiving object MUST be prepared  
761 to receive all supported attributes. The Job object can later be queried to find out what Job  
762 Template attributes were originally requested in the create request, and such attributes are  
763 returned in the response as Job Object Attributes. The Printer object can be queried about its Job  
764 Template attributes to find out what type of job processing capabilities are supported and/or what  
765 the default job processing behaviors are, though such attributes are returned in the response as  
766 Printer Object Attributes. The "ipp-attribute-fidelity" operation attribute affects processing of all  
767 client supplied Job Template attributes (see section 16 for a full description of "ipp-attribute-  
768 fidelity" and its relationship to other attributes).
- 769 - Job Object Attributes: These attributes are returned in response to a query operation directed at a  
770 Job object.
- 771 - Printer Object Attributes: These attributes are returned in response to a query operation directed at  
772 a Printer object.
- 773 - Unsupported Attributes: In a create request, the client supplies a set of Operation and Job Template  
774 attributes. If any of these attributes or their values are unsupported by the Printer object, the  
775 Printer object returns the set of unsupported attributes in the response. Section 16 gives a full  
776 description of how Job Template attributes supplied by the client in a create request are processed



777 by the Printer object and how unsupported attributes are returned to the client. Because of  
778 extensibility, any IPP object might receive a request that contains new ~~(or for any reason or~~  
779 ~~unknown)~~ attributes or values ~~for which it has no support. that it does not support.~~ In such cases  
780 ~~for any operation request (not just create requests),~~ the IPP object processes what it can and  
781 ~~MUST~~ returns the ~~se~~ unsupported attributes in the response.

782

783 Later in this section, each operation is formally defined by identifying the allowed and expected groups of  
784 attributes for each request and response. The model identifies a specific order for each group in each  
785 request or response, but the attributes within each group may be in any order, unless specified otherwise.

786 Each attribute specification includes the attribute's name followed by the name of its attribute syntax(es)  
787 in parentheses. In addition, ~~the each~~ 'integer' attributes ~~are is~~ followed by the allowed range in  
788 parentheses, (m:n), for ~~values of that attribute. the integer value.~~ ~~The Each~~ 'text' ~~and or~~ 'name'  
789 attributes ~~are is~~ followed by the maximum size in octets in parentheses, (size), for values of that attribute.  
790 ~~in octets in parentheses.~~ For more details on attribute syntax notation, see the descriptions of these  
791 attributes syntaxes in section 4.1. It is an operation al error for clients to supply in operation requests  
792 and/or IPP objects to returns in operations responses attribute value(s) that do not match the syntax(es)  
793 defined for that attribute (see section 3 for operation attributes and section 4 for IPP object attributes).

794 Note: Document data included in the operation is not strictly an attribute, but it is treated as a special  
795 attribute group for ordering purposes. The only operations that support supplying the document data  
796 within an operation request are Print-Job and Send-Document. There are no operation responses that  
797 include document data.

798 Note: Some operations are MANDATORY for IPP objects to support; the others are OPTIONAL (see  
799 section 5.2.2). Therefore, before using an OPTIONAL operation, a client SHOULD first use the  
800 MANDATORY Get-Printer-Attributes operation to query the Printer's "operations-supported" attribute  
801 in order to determine which OPTIONAL Printer and Job operations are actually supported. The client  
802 SHOULD NOT use an OPTIONAL operation that is not supported. When an IPP object receives a  
803 request to perform an operation it does not support, it returns the 'server-error-operation-not-supported'  
804 status code (see section 14.1.5.2). An IPP object is non-conformant if it does not support a  
805 MANDATORY operation.

#### 806 3.1.33.1.4 Character Set and Natural Language Operation Attributes

807 Some Job and Printer attributes have values that are text strings and names intended for human  
808 understanding rather than machine understanding (see the 'text' and 'name' attribute syntax descriptions in  
809 section 4.1). The following sections describe two special Operation Attributes called "attributes-charset"  
810 and "attributes-natural-language". These attributes are always part of the Operation Attributes group.  
811 For most attribute groups, the order of the attributes within the group is not important. However, for  
812 these two attributes within the Operation Attributes group, the order is critical. The "attributes-charset"

813 attribute MUST be the first attribute in the group and the "attributes-natural-language" attribute MUST  
814 be the second attribute in the group. In other words, these attributes MUST be supplied in every IPP  
815 request and response, they MUST come first in the group, and MUST come in the specified order. For  
816 job creation operations, the IPP Printer implementation saves these two attributes with the new Job  
817 object as Job Description attributes. For the sake of brevity in this document, these operation attribute  
818 descriptions are not repeated with every operation request and response, but have a reference back to this  
819 section instead.

#### 820 3.1.3.13.1.4.1 Request Operation Attributes

821 The client SHALL supply and the Printer object SHALL support the following MANDATORY operation  
822 attributes in every IPP/1.0 operation request:

823 "attributes-charset" (charset):

824 This operation attribute identifies the charset (coded character set and encoding method) used by  
825 any 'text' and 'name' attributes that the client is supplying in this request. It also identifies the  
826 charset that the Printer object SHALL use (if supported) for all 'text' and 'name' attributes and  
827 status messages that the Printer object returns in the response to this request. See Sections 4.1.1  
828 and 4.1.24.1.2 for the specification of the 'text' and 'name' attribute syntaxes.

829  
830 All IPP objects SHALL support the 'utf-8' charset [RFC2044] and MAY support additional  
831 charsets provided that they are registered with IANA [IANA-CS]. If the Printer object does not  
832 support the client supplied charset value, the Printer object SHALL reject the request and return  
833 the 'client-error-charset-not-supported' status code. The Printer object SHALL indicate the  
834 charset(s) supported as the values of the "charset-supported" Printer attribute (see Section  
835 4.4.15), so that the client MAY-can query to determine which charset(s) are supported.

836  
837 Note to client implementers: Since IPP objects are only required to support the 'utf-8' charset, in  
838 order to maximize interoperability with multiple IPP object implementations, a client may want to  
839 supply 'utf-8' in the "attributes-charset" operation attribute, even though the client is only passing  
840 and able to present a simpler charset, such as US-ASCII or ISO-8859-1. Then the client will have  
841 to filter out (or charset convert) those characters that are returned in the response that it cannot  
842 present to its user. On the other hand, if both the client and the IPP objects also support a charset  
843 in common besides utf-8, the client MAY-may want to use that charset in order to avoid charset  
844 conversion or data loss.

845  
846 See the 'charset' attribute syntax description in Section 4.1.7 for the syntax and semantic  
847 interpretation of the values of this attribute and for example values.

848



849 "attributes-natural-language" (naturalLanguage):

850 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that  
851 the client is supplying in this request. This attribute also identifies the natural language that the  
852 Printer object SHOULD use for all 'text' and 'name' attributes and status messages that the Printer  
853 object returns in the response to this request.

854  
855 There are no MANDATORY natural languages required for the Printer object to support.  
856 However, the Printer object's "generated-natural-language-supported" attribute ~~SHALL identify~~  
857 identifies the natural languages supported by the Printer object and any contained Job objects for  
858 all text strings generated by the IPP object. A client MAY query this attribute to determine which  
859 natural language(s) are supported for generated messages.

860  
861 For any of the attributes for which the Printer object generates text, i.e., for the "job-state-  
862 message", "printer-state-message", and status messages (see Section 3.1.63.1.4), the Printer  
863 object SHALL be able to generate these text strings in any of its supported natural languages. If  
864 the client requests a natural language that is not supported, the Printer object SHALL return these  
865 generated messages in the Printer's configured natural language as specified by the Printer's  
866 "natural-language-configured" attribute" (see Section 4.4.16).

867  
868 For other 'text' and 'name' attributes supplied by the client, authentication system, operator,  
869 system administrator, or manufacturer, i.e., for "job-originating-user-name", "printer-name"  
870 (name), "printer-location" (text), "printer-info" (text), and "printer-make-and-model" (text), the  
871 Printer object is only required to support the configured natural language of the Printer identified  
872 by the Printer object's "natural-language-configured" attribute, though support of additional  
873 natural languages for these attributes is permitted.

874  
875 For any 'text' or 'name' attribute in the request that is in a different natural language than the value  
876 supplied in the "attributes-natural-language", the client SHALL use the Natural Language  
877 Override mechanism (see sections 4.1.1.2 and 4.1.2.2) for each such attribute value supplied.

878  
879 The IPP object SHALL accept any natural language and any Natural Language Override, whether  
880 the IPP object supports that natural language or not (and independent of the value of the "ipp-  
881 attribute-fidelity" Operation attribute). That is the IPP object accepts all client supplied values no  
882 matter what the values are in the Printer object's "generated-natural-language-supported"  
883 attribute. That attribute, "generated-natural-language-supported", only applies to generated  
884 messages, not client supplied messages. The IPP object SHALL remember that natural language  
885 for all client supplied attributes, and when returning those attributes in response to a query, the  
886 IPP object SHALL indicate that natural language.

887

888 For example, the "job-name" attribute MAY be supplied by the client in a create request. The text  
889 value for this attribute will be in the natural language identified by the "attribute-natural-language"  
890 attribute, or if different, as identified by the Natural Language Override mechanism. If supplied,  
891 the IPP object will use the value of the "job-name" attribute to populate the Job object's "job-  
892 name" attribute. Whenever any client queries the Job object's "job-name" attribute, the IPP object  
893 returns the attribute as stored and uses the Natural Language Override mechanism to specify the  
894 natural language, if it is different from that reported in the "attributes-natural-language" operation  
895 attribute of the response. An IPP object SHALL NOT reject a request based on a supplied  
896 natural language in an "attributes-natural-language" Operation attribute or in any attribute that  
897 uses the Natural Language Override.

898  
899 See the 'naturalLanguage' attribute syntax description in section 4.1.8 for the syntax and semantic  
900 interpretation of the values of this attribute and for example values.

901  
902 Clients SHOULD NOT supply 'text' or 'name' attributes that request-use an illegal combination of natural  
903 language and charset. For example, suppose a Printer object supports charsets 'utf-8', 'iso-8859-1', and  
904 'iso-8859-7'. Suppose it also supports natural languages 'en' (English), 'fr' (French), and 'el' (Greek).  
905 Although the Printer object supports the charset 'iso-8859-1' and natural language 'el', it probably does  
906 not support the combination of Greek text strings using the 'iso-8859-1' charset. In a create request, if a  
907 client supplies a "job-name" operation attribute that uses that specific invalid combination, it is a client  
908 choice and it doesn't affect the Printer object or its correct operation to accept the invalid combination.  
909 In this case, the Printer object simply accepts the client supplied value, stores it with the Job object, and  
910 responds back with the same invalid combination whenever any client queries for that attribute. In a  
911 query type operation (Get-Printer-Attributes for example), if the client requests an invalid combination,  
912 the Printer object simply responds (as described below) using the Printer's configured natural language  
913 rather than the natural language requested by the client. In either case, the Printer object does not reject  
914 the request because of an invalid combination of charset and natural language (either at the global  
915 operation level or at the Natural Language Override attribute-by-attribute level). If the client requests  
916 'iso-8859-1' and 'el', it is an invalid combination of charset and natural language. In this case, the IPP  
917 object SHALL NOT change either of these attribute values and SHALL accept them as if they were  
918 valid.

#### 919 3.1.3.23.1.4.2 Response Operation Attributes

920 The Printer object SHALL supply and the client SHALL support the following MANDATORY operation  
921 attributes in every IPP/1.0 operation response:

922 "attributes-charset" (charset):

923 This operation attribute identifies the charset used by any 'text' and 'name' attributes that the  
924 Printer object is returning in this response. The value in this response SHALL be the same value

925 as the "attributes-charset" operation attribute supplied by the client in the request. If this is not  
926 possible (i.e., the charset requested is not supported), the request ~~SHALL~~would have been  
927 rejected. See "attributes-charset" described in Section 3.1.4.1 above.

928  
929 If the Printer object supports more than just the 'utf-8' charset, the Printer object SHALL be able  
930 to code convert between each of the charsets supported on a highest fidelity possible basis in  
931 order to return the 'text' and 'name' attributes in the charset requested by the client. However,  
932 some information loss MAY occur during the charset conversion depending on the charsets  
933 involved. For example, the Printer object may convert from a UTF-8 'a' to a US-ASCII 'a' (with  
934 no loss of information), from an ISO Latin 1 CAPITAL LETTER A WITH ACUTE ACCENT  
935 to US-ASCII 'A' (losing the accent), or from a UTF-8 Japanese Kanji character to some ISO  
936 Latin 1 error character indication such as '?', decimal code equivalent, or to the absence of a  
937 character, depending on implementation.

938  
939 Note: Whether an implementation that supports more than one charset stores the data in the  
940 charset supplied by the client or code converts to one of the other supported charsets, depends on  
941 implementation. The strategy ~~SHOULD~~should try to minimize loss of information during code  
942 conversion. On each response, such an implementation converts from its internal charset to that  
943 requested.

944  
945 "attributes-natural-language" (naturalLanguage):

946 This operation attribute identifies the natural language used by any 'text' and 'name' attributes that  
947 the IPP object is returning in this response. Unlike the "attributes-charset" operation attribute, the  
948 IPP object NEED NOT return the same value as that supplied by the client in the request. The  
949 IPP object MAY return the natural language of the Job object or the Printer's configured natural  
950 language as identified by the Printer object's "natural-language-configured" attribute, rather than  
951 the natural language supplied by the client. For any 'text' or 'name' attribute or status message in  
952 the response that is in a different natural language than the value returned in the "attributes-  
953 natural-language" operation attribute, the IPP object SHALL use the Natural Language Override  
954 mechanism (see sections 4.1.1.2 and 4.1.2.2) on each attribute value returned.

### 955 ~~3.1.43.1.5~~ Operation Targets

956 All IPP operations are directed at IPP objects. For Printer operations, the operation is always directed at  
957 a Printer object using one of its URIs (i.e., one of the values in the Printer object's "printer-uri-supported"  
958 attribute). Even if the Printer object supports more than one URI, the client supplies only one URI as the  
959 target of the operation. The client identifies the target object by supplying the correct URI in the  
960 "printer-uri (uri)" operation attribute.

961 For Job operations, the operation is directed at either:

- 962 - The Job object itself using the Job object's URI. In this case, the client identifies the target object by  
963 supplying the correct URI in the "job-uri (uri)" operation attribute.
- 964 - The Printer object that created the Job object using both the Printer objects URI and the Job object's  
965 Job ID. Since the Printer object that created the Job object generated the Job ID, it MUST be  
966 able to correctly associate the client supplied Job ID with the correct Job object. The client  
967 supplies the Printer object's URI in the "printer-uri (uri)" operation attribute and the Job object's  
968 Job ID in the "job-id (integer(1:MAX))" operation attribute.

969

970 If the operation is directed at the Job object directly using the Job object's URI, the client SHALL NOT  
971 include the redundant "job-id" operation attribute.

972 The operation target attributes are MANDATORY operation attributes that MUST be included in every  
973 operation request. Like the charset and natural language attributes (see section 3.1.4), the operation  
974 target attributes are specially ordered operation attributes. In all cases, the operation target attributes  
975 immediately follow the "attributes-charset" and "attributes-natural-language" attributes within the  
976 operation attribute group, however the specific ordering rules are :

- 977 - In the case where there is only one operation target attribute (i.e., either only the "printer-uri"  
978 attribute or only the "job-uri" attribute), that attribute MUST be the third attribute in the  
979 operation attributes group.
- 980 - In the case where Job operations use two operation target attributes (i.e., the "printer-uri" and "job-  
981 id" attributes), the "printer-uri" attribute MUST be the third attribute and the "job-id" attribute  
982 MUST be the fourth attribute.

983

984 Note: The IPP transport and encoding document [IPP-PRO] calls for the target URL to be included both  
985 inside the IPP operation (as MANDATORY operation attributes) and outside the operation (at the HTTP  
986 layer). The potential exists that these two values reference the same IPP object, but are not literally  
987 identical since one can be a relative URL and the other can be an absolute URL. HTTP/1.1 allows clients  
988 to generate and send a relative URL rather than an absolute URL. A relative URL identifies a resource  
989 with the scope of the HTTP server, but does not include scheme, host or port. The following statements  
990 characterize how URLs should be used in the mapping of IPP onto HTTP/1.1:

- 991 1. Although potentially redundant, a client MUST supply the target of the operation both as an  
992 Operation Attribute (see Section 3.1.5) and as a URL at the HTTP layer. The rationale for this  
993 decision is to maintain a consistent set of rules for mapping IPP to possibly many communication  
994 layers, even where URLs are not used as the addressing mechanism.
- 995 2. Even though these two URLs might not be literally identical (one being relative and the other being  
996 absolute), they must both reference the same IPP object.
- 997 3. The URL in the HTTP layer is either relative or absolute and is used by the HTTP server to route  
998 the HTTP request to the correct resource relative to that HTTP server. The HTTP server need  
999 not be aware of the URL within the operation request.

1000 4. Once the HTTP server resource begins to process the HTTP request, it might get the reference to  
1001 the appropriate IPP Printer object from either the HTTP URL (using to the context of the HTTP  
1002 server for relative URLs) or from the URL within the operation request; the choice is up to the  
1003 implementation.

1004 5. HTTP URLs can be relative or absolute, but the target URL in the operation MUST be an absolute  
1005 URL

1006  
1007 The following rules apply to the use of port numbers in URIs that identify IPP objects:

1008 1. If ~~the URI scheme the protocol scheme for the URI~~ allows the port number to be explicitly  
1009 included in the URI string, and an ~~explicit~~ port number is specified within ~~the syntax of~~ the URI,  
1010 then that port number MUST be used by the client to contact the IPP object.

1011  
1012 2. If the URI scheme allows the port number to be explicitly included in the URI string, and a port  
1013 number is not specified within the URI, then default port number implied by that URI MUST be  
1014 used by the client to contact the IPP object.

1015  
1016 3. If the ~~protocol~~ \_\_\_\_\_ URI scheme ~~for the URI~~ does not allow an explicit port number  
1017 specification to be specified within the URI, then the default port number implied by that URI ~~for~~  
1018 the ~~protocol~~ MUST be used by the client to contact the IPP object.

1019  
1020 Note: The IPP transport and encoding document [IPP-PRO] shows a mapping of IPP onto HTTP/1.1  
1021 and defines a new default port number for using IPP over HTTP/1.1.

### 1022 1023 3.1.53.1.6 Operation Status Codes and Messages

1024 ~~Every~~ Every operation response ~~includes a returns a~~ MANDATORY "status-code" ~~operation attribute~~  
1025 and an OPTIONAL "status-message" operation attribute. ~~The "status-code" attribute~~ provides  
1026 information on the processing of a request. A "status-message" attribute provides a short textual  
1027 description of the status of the operation. The status code is intended for use by automata, and the status  
1028 message is intended for the human end user. If a response does include a "status-message" attribute, an  
1029 IPP client NEED NOT examine or display the message, however it SHOULD do so in some  
1030 implementation specific manner.

1031 The ~~value of a~~ "status-code" ~~value attribute~~ is a numeric value that has semantic meaning. The "status-  
1032 code" syntax is similar to a "type2 enum" (see section 4.1 on "Attribute Syntaxes") except that values can  
1033 range only from 0x0000 to 0x7FFF. Section 14 describes the status codes, assigns the numeric values,  
1034 and suggests a corresponding status message for each status code. The "status-message" attribute's  
1035 syntax is "text(255)".

1036 A client implementation of IPP SHOULD convert status code values into any localized message that has  
1037 semantic meaning to the end user. If the Printer object supports the status message, the Printer object  
1038 MUST be able to generate this message in any of the natural languages identified by the Printer object's  
1039 "generated-natural-language-supported" attribute (see the "attributes-natural-language" operation  
1040 attribute specified in section 3.1.4.1). As described in section 3.1.4.1 for any returned 'text' attribute, if  
1041 there is a choice for generating this message, the Printer object uses the natural language indicated by the  
1042 value of the "attributes-natural-language" in the client request if supported, otherwise the Printer object  
1043 uses the value in the Printer object's own "natural-language-configured" attribute.

#### 1044 3.1.63.1.7 Versions

1045 Each operation request and response carries with it a "version-number"-~~attribute~~. Each value of the  
1046 "version-number" ~~attribute~~ is in the form "X.Y" where X is the major version number and Y is the minor  
1047 version number. By including a version number in the client request, it allows the client (~~the requester~~) to  
1048 identify which version of IPP it is interested in using. If the IPP object does not support that version, the  
1049 object responds with a status code of 'server-error-version-not-supported'.

1050 There is no version negotiation per se. However, if after receiving a 'server-error-version-not-supported'  
1051 status code from an IPP object, there is nothing that prevents a client from trying again with a different  
1052 version number. In order to conform to IPP/1.0, an implementation MUST support at least version '1.0'.

1053 There is only one notion of "version number" that covers both IPP Model and IPP Protocol changes.  
1054 Thus the version number MUST change when introducing a new version of the Model document or a  
1055 new version of the Protocol document.

1056 Changes to the major version number indicate structural or syntactic changes that make it impossible for  
1057 older version of IPP clients and Printer objects to correctly parse and process the new or changed  
1058 attributes, operations and responses. If the major version number changes, the minor version numbers is  
1059 set to zero. As an example, adding the "ipp-attribute-fidelity" attribute (if it had not been part of version  
1060 '1.0'), would have required a change to the major version number. Items that might affect the changing of  
1061 the major version number include any changes to the protocol specification itself, such as:

- 1062 - reordering of ordered attributes or attribute sets
- 1063 - changes to the syntax of existing attributes
- 1064 - changing Operation or Job Template attributes from OPTIONAL to MANDATORY and vice versa
- 1065 - adding MANDATORY (for an IPP object to support) operation attributes
- 1066 - adding MANDATORY (for an IPP object to support) operation attribute groups
- 1067 - adding values to existing operation attributes
- 1068 - adding MANDATORY operations

1069



1070 Changes to the minor version number indicate the addition of new features, attributes and attribute values  
1071 that may not be understood by all IPP objects, but which can be ignored if not understood. Items that  
1072 might affect the changing of the minor version number include any changes to the model objects and  
1073 attributes but not the protocol specification itself (except adding attribute syntaxes), such as:

- 1074 - grouping all extensions not included in a previous version into a new version
- 1075 - adding new attribute values
- 1076 - adding new object attributes
- 1077 - adding OPTIONAL (for an IPP object to support) operation attributes (i.e., those attributes that an  
1078 IPP object can ignore without confusing clients)
- 1079 - adding OPTIONAL (for an IPP object to support) operation attribute groups (i.e., those attributes  
1080 that an IPP object can ignore without confusing clients)
- 1081 - adding new attribute syntaxes
- 1082 - adding OPTIONAL operations
- 1083 - changing Job Description attributes or Printer Description attributes from OPTIONAL to  
1084 MANDATORY or vice versa.

1085  
1086 The encoding of the "operation-id", the "version-number", the "status-code", and the "request-id"  
1087 SHALL NOT change over any version number (either major or minor). This rule guarantees that all  
1088 future versions will be backwards compatible with all previous versions (at least for checking the  
1089 "operation-id", the "version-number", and the "request-id"). In addition, any protocol elements  
1090 (attributes, error codes, tags, etc.) that are not carried forward from one version to the next are  
1091 deprecated so that they can never be reused with new semantics.

1092 Implementations that support a certain major version NEED NOT support ALL previous versions. As  
1093 each new major version is defined (through the release of a new specification), that major version will  
1094 specify which previous major versions MUST be supported in compliant implementations.

#### 1095 3.1.73.1.8 Job Creation Operations

1096 In order to "submit a print job" and create a new Job object, a client issues a create request. A create  
1097 request is any one of following three operation requests:

- 1098 - The Print-Job Request: A client that wants to submit a print job with only a single document uses  
1099 the Print-Job operation. The operation allows for the client to "push" the document data to the  
1100 Printer object by including the document data in the request itself.
- 1101  
1102 - The Print-URI Request: A client that wants to submit a print job with only a single document  
1103 (where the Printer object "pulls" the document data instead of the client "pushing" the data to the  
1104 Printer object) uses the Print-URI operation. In this case, the client includes in the request only a  
1105 URI reference to the document data (not the document data itself).

1106  
1107 - The Create-Job Request: A client that wants to submit a print job with multiple documents uses the  
1108 Create-Job operation. This operation is followed by an arbitrary number of Send-Document  
1109 and/or Send-URI operations (each creating another document for the newly create Job object).  
1110 The Send-Document operation includes the document data in the request (the client "pushes" the  
1111 document data to the printer), and the Send-URI operation includes only a URI reference to the  
1112 document data in the request (the Printer "pulls" the document data from the referenced location).  
1113 The last Send-Document or Send-URI request for a given Job object includes a "last-document"  
1114 operation attribute set to 'true' indicating that this is the last request.

1115  
1116 Throughout this model specification, the term "create request" is used to refer to any of these three  
1117 operation requests.

1118 A Create-Job operation followed by only one Send-Document operation is semantically equivalent to a  
1119 Print-Job operation, however, for performance reasons, the client SHOULD use the Print-Job operation  
1120 for all single ~~Document~~document Jobs~~jobs~~. Also, Print-Job is a MANDATORY operation (all  
1121 implementations MUST support it) whereas Create-Job is an OPTIONAL operation, hence some  
1122 implementations might not support it.

1123 Job submission time is the point in time when a client issues a create request. The initial state of every  
1124 Job object is the 'pending' or 'pending-held' state. Later, the Printer object begins processing the print job.  
1125 At this point in time, the Job object's state moves to 'processing'. This is known as job processing time.  
1126 There are validation checks that must be done at job submission time and others that must be performed  
1127 at job processing time.

1128 At job submission time and at the time a Validate-Job operation is received, the Printer MUST do the  
1129 following:

- 1130 1. Process the client supplied attributes and either accept or reject the request  
1131 2. Validate the syntax of and support for the scheme of any client supplied URI

1132  
1133 Section 16 describes the rules and issues surrounding the processing of client supplied attributes. Section  
1134 16.3 presents suggested steps for an IPP object to either accept or reject any request. Section 16.4  
1135 presents suggested additional steps for processing create requests.

1136 At job submission time the Printer SHOULD NOT perform the validation checks reserved for job  
1137 processing time such as:

- 1138 1. ~~Validate~~Validating the document data  
1139 2. ~~Validate~~Validating the actual contents of any client supplied URI (resolve the reference and follow  
1140 the link to the document data)



1141

1142 At job submission time, these additional job processing time validation checks are essentially useless,  
1143 since they require actually parsing and interpreting the document data, are not guaranteed to be 100%  
1144 accurate, and MUST ~~yet~~ be done, yet again, at job processing time. Also, in the case of a URI, checking  
1145 for availability at job submission time does not guarantee availability at job processing time. In addition,  
1146 at job processing time, the Printer object might discover any of the following conditions that were not  
1147 detectable at job submission time:

- 1148 - runtime errors in the document data,
- 1149 - nested document data that is in an unsupported format,
- 1150 - the URI reference is no longer valid (i.e., the server hosting the document might be down), or
- 1151 - any other job processing error

1152

1153 At job processing time, since the Printer object has already responded with a successful status code in the  
1154 response to the create request, if the Printer object detects an error, the Printer object is unable to inform  
1155 the end user of the error with an operation status code. In this case, the Printer, depending on the error,  
1156 can set the "job-state", "job-state-reasons", or "job-state-message" attributes to the appropriate value(s)  
1157 so that later queries can report the correct job status.

1158 Note: Asynchronous notification of events is outside the scope of IPP/1.0.

## 1159 3.2 Printer Operations

1160 All Printer operations are directed at Printer objects. A client MUST always supply the "printer-uri"  
1161 operation attribute in order to identify the correct target of the operation.

### 1162 3.2.1 Print-Job Operation

1163 This MANDATORY operation allows a client to submit a print job with only one document and supply  
1164 the document data (rather than just a reference to the data). See Section 16 for the suggested steps for  
1165 processing create operations and their Operation and Job Template attributes.

#### 1166 3.2.1.1 Print-Job Request

1167 The following groups of attributes are supplied as part of the Print-Job Request:

1168 Group 1: Operation Attributes

1169 Natural Language and Character Set:

1170 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1171 3.1.4.1. The Printer object SHALL copy these values to the corresponding Job Description  
1172 attributes described in sections 4.3.23 and 4.3.24.

1173

1174 Target:

1175 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1176 section 3.1.5.

1177

1178 Requesting User Name:

1179 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1180 described in section 8.3.

1181

1182 "job-name" (name(MAX)):

1183 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1184 attribute. It contains the client supplied Job name. If this attribute is supplied by the client, its  
1185 value is used for the "job-name" attribute of the newly created Job object. The client MAY  
1186 automatically include any information that will help the end-user distinguish amongst his/her jobs,  
1187 such as the name of the application program along with information from the document, such as  
1188 the document name, document subject, or source file name. If this attribute is not supplied by the  
1189 client, the Printer generates a name to use in the "job-name" attribute of the newly created Job  
1190 object (see Section 4.3.5).

1191

1192 "ipp-attribute-fidelity" (boolean):

1193 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1194 attribute. The value 'true' indicates that total fidelity to client supplied Job Template attributes  
1195 and values is required, else the Printer object SHALL reject the Print-Job request. The value  
1196 'false' indicates that a reasonable attempt to print the Job object is acceptable and the Printer  
1197 object SHALL accept the Print-job request. If not supplied, the Printer object assumes the value is  
1198 'false'. All Printer objects MUST support both types of job processing. See section 16 for a full  
1199 description of "ipp-attribute-fidelity" and its relationship to other attributes, especially the Printer  
1200 object's "pdl-override-supported" attribute.

1201

1202 "document-name" (name(MAX)):

1203 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1204 attribute. It contains the client supplied document name. The document name MAY be different  
1205 than the Job name. Typically, the client software automatically supplies the document name on  
1206 behalf of the end user by using a file name or an application generated name. If this attribute is  
1207 supplied, its value can be used in a manner defined by each implementation. Examples include:  
1208 printed along with the Job (job start sheet, page adornments, etc.), used by accounting or

1209 resource tracking management tools, or even stored along with the document as a document level  
1210 attribute. IPP/1.0 does not support the concept of document level attributes.

1211

1212 "document-format" (mimeMediaType) :

1213 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1214 attribute. The value of this attribute identifies the format of the supplied document data. If the  
1215 client does not supply this attribute, the Printer object assumes that the document data is in the  
1216 format defined by the Printer object's "document-format-default" attribute. If the client supplies  
1217 this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the  
1218 values of the Printer object's "document-format-supported" attribute, the Printer object SHALL  
1219 reject the request and return the 'client-error-document-format-not-supported' status code.

1220

1221 "document-natural-language" (naturalLanguage):

1222 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1223 this attribute. This attribute specifies the natural language of the document for those document-  
1224 formats that require a specification of the natural language in order to image the document  
1225 unambiguously. There are no particular values required for the Printer object to support.

1226

1227 "compression" (type3 keyword)

1228 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1229 this attribute and the "compression-supported" attribute (see section 4.4.29). The client supplied  
1230 "compression" operation attribute identifies the compression algorithm used on the document  
1231 data. If the client omits this attribute, the Printer object SHALL assume that the data is not  
1232 compressed. If the client supplies the attribute and the Printer object supports the attribute, the  
1233 Printer object uses the corresponding decompression algorithm on the document data. If the client  
1234 supplies this attribute, but the value is not supported by the Printer object, i.e., the value is not  
1235 one of the values of the Printer object's "compression-supported" attribute, the Printer object  
1236 SHALL copy the attribute and its value to the Unsupported Attributes response group, reject the  
1237 request, and return the 'client-error-attributes-or-values-not-supported' status code.

1238

1239 "job-k-octets" (integer(0:MAX))

1240 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1241 this attribute and the "job-k-octets-supported" attribute (see section 4.4.30). The client supplied  
1242 "job-k-octets" operation attribute identifies the total size of the document(s) in K octets being  
1243 submitted (see section 4.3.17 for the complete semantics). If the client supplies the attribute and  
1244 the Printer object supports the attribute, the value of the attribute is used to populate the Job  
1245 object's "job-k-octets" Job Description attribute.

1246

1247 Note: For this attribute and the following two attributes ("job-impressions", and "job-media-  
1248 sheets"), if the client supplies the attribute, but the Printer object does not support the attribute,

1249 the Printer object ignores the client-supplied value. If the client supplies the attribute and the  
1250 Printer supports the attribute, and the value is within the range of the corresponding Printer  
1251 object's "xxx-supported" attribute, the Printer object SHALL use the value to populate the Job  
1252 object's "xxx" attribute. If the client supplies the attribute and the Printer supports the attribute,  
1253 but the value is outside the range of the corresponding Printer object's "xxx-supported" attribute,  
1254 the Printer object SHALL copy the attribute and its value to the Unsupported Attributes response  
1255 group, reject the request, and return the 'client-error-attributes-or-values-not-supported' status  
1256 code. If the client does not supply the attribute, the Printer object MAY choose to populate the  
1257 corresponding Job object attribute depending on whether the Printer object supports the attribute  
1258 and is able to calculate or discern the correct value.

1259

1260 "job-impressions" (integer(0:MAX))

1261 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1262 this attribute and the "job-impressions-supported" attribute (see section 4.4.31). The client  
1263 supplied "job-impressions" operation attribute identifies the total size in number of impressions of  
1264 the document(s) being submitted (see section 4.3.18 for the complete semantics).

1265

1266 See note under "job-k-octets".

1267

1268 "job-media-sheets" (integer(0:MAX))

1269 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1270 this attribute and the "job-media-sheets-supported" attribute (see section 4.4.32). The client  
1271 supplied "job-media-sheets" operation attribute identifies the total number of media sheets to be  
1272 produced for this job (see section 4.3.19 for the complete semantics).

1273

1274 See note under "job-k-octets".

1275

1276 Group 2: Job Template Attributes

1277 The client OPTIONALLY supplies a set of Job Template attributes as defined in section 4.2.

1278

1279 Group 3: Document Content

1280 The client MUST supply the document data to be processed.

1281

1282 Note: In addition to the MANDTORY common elements required for every operation request, The the  
1283 simplest Print-Job Request consists of just the "attributes-charset" and "attributes-natural-language"  
1284 operation attributes; the "printer-uri" target operation attribute; the Document Content and -and nothing  
1285 else. In this simple case, the Printer object:

1286 - creates a new Job object (the Job object contains a single document),

- 1287 - stores a generated Job name in the "job-name" attribute in the natural language and charset  
1288 requested (see Section 3.1.4.1) (if those are supported, otherwise using the Printer object's default  
1289 natural language and charset), and  
1290 - at job processing time, uses its corresponding default value attributes for the supported Job  
1291 Template attributes that were not supplied by the client as IPP attribute or embedded instructions  
1292 in the document data.  
1293

### 1294 3.2.1.2 Print-Job Response

1295 The Printer object SHALL return to the client the following sets of attributes as part of the Print-Job  
1296 Response:

#### 1297 Group 1: Operation Attributes

1298 Status ~~Code and~~ Message:

1299 ~~The response includes the MANDATORY status code and an OPTIONAL~~ In addition to the  
1300 ~~MANDATORY status code returned in every response, the response~~ OPTIONALLY includes a  
1301 "status-message" (text) operation attribute as described in section 3.1.6~~3.1.4~~. If the client  
1302 supplies unsupported or conflicting Job Template attributes or values, the Printer object SHALL  
1303 reject or accept the Print-Job request depending on the whether the client supplied a 'true' or  
1304 'false' value for the "ipp-attribute-fidelity" operation attribute. See section 16 for a complete  
1305 description of the suggested steps for processing a create request.  
1306

1307 Natural Language and Character Set:

1308 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1309 3.1.4.2.  
1310

#### 1311 Group 2: Unsupported Attributes

1312 This is a set of Operation and Job Template attributes supplied by the client (in the request) that  
1313 are not supported by the Printer object or that conflict with one another (see sections 16.3 and  
1314 16.4).  
1315

1316 Unsupported attributes fall into three categories:

- 1317
- 1318 1. The Printer object does not support the named attribute (no matter what the value).
  - 1319 2. The Printer object does support the attribute, but does not support some or all of the particular  
1320 values supplied by the client (i.e., the Printer object does not have those values in the  
1321 corresponding supported values attribute).

1322 3. The Printer object does support the attributes and values supplied, but the particular values are  
1323 in conflict with one another, because they violate a constraint, such as not being able to  
1324 staple transparencies.

1325  
1326 In the case of an unsupported attribute name, the Printer object returns the client-supplied  
1327 attribute with a substituted "out-of-band" value of 'unsupported' indicating no support for the  
1328 attribute itself (see the beginning of section 4.1).

1329  
1330 In the case of a supported attribute with one or more unsupported values, the Printer object  
1331 simply returns the client-supplied attribute with the unsupported values as supplied by the client.  
1332 This indicates support for the attribute, but no support for that particular value. If the client  
1333 supplies a multi-valued attribute with more than one value and the Printer object supports the  
1334 attribute but only supports a subset of the client supplied values, the Printer object SHALL return  
1335 only those values that are unsupported.

1336  
1337 In the case of two (or more) supported attribute values that are in conflict with one another  
1338 ~~(although supported they values conflict when -requested within the same job)because they cannot~~  
1339 ~~be used together in the same job~~, the Printer object SHALL return all the values that it ignores or  
1340 substitutes to resolve the conflict, but not any of the values that it is still using. The choice for  
1341 exactly how to resolve the conflict is implementation dependent. See Section 16.4.4 for an  
1342 example.

1343  
1344 In these three cases, the value of the "ipp-attribute-fidelity" supplied by the client does not affect  
1345 what the Printer object returns. The value of "ipp-attribute-fidelity" only affects whether the  
1346 Print-Job operation is accepted or rejected. If the job is accepted, the client may query the job  
1347 using the Get-Job-Attributes operation requesting the unsupported attributes that were returned in  
1348 the create response to see which attributes were ignored (not stored on the Job object) and which  
1349 attributes were stored with other (substituted) values.

1350

1351 Group 3: Job Object Attributes

1352 "job-uri" (uri):

1353 The Printer object MUST return the Job object's URI by returning the contents of the  
1354 MANDATORY "job-uri" Job object attribute. The client uses the Job object's URI when  
1355 directing operations at the Job object. The Printer object always uses its configured security  
1356 policy when creating the new URI. However, if the Printer object supports more than one URI,  
1357 the Printer object also uses information about which URI was used in the Print-Job Request to  
1358 generated the new URI so that the new URI references the correct access channel. In other  
1359 words, if the Print-Job Request comes in over a secure channel, the Printer object MUST generate  
1360 a Job URI that uses the secure channel as well.

1361

1362 "job-id" (integer(1:MAX)):

1363 The Printer object MUST return the Job object's Job ID by returning the MANDATORY "job-  
1364 id" Job object attribute. The client uses this "job-id" attribute in conjunction with the "printer-uri"  
1365 attribute used in the Print-Job Request when directing Job operations at the Printer object.

1366

1367 "job-state":

1368 The Printer object MUST return the Job object's MANDATORY "job-state" attribute. The value  
1369 of this attribute (along with the value of the next attribute "job-state-reasons") is taken from a  
1370 "snapshot" of the new Job object at some meaningful point in time (implementation defined)  
1371 between when the Printer object receives the Print-Job Request and when the Printer object  
1372 returns the response.

1373

1374 "job-state-reasons":

1375 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-reasons"  
1376 attribute. If the Printer object supports this attribute then it MUST be returned in the response.  
1377 If this attribute is not returned in the response, the client can assume that the "job-state-reasons"  
1378 attribute is not supported and will not be returned in a subsequent Job object query.

1379

1380 "job-state-message":

1381 The Printer object OPTIONALLY returns the Job object's OPTIONAL "job-state-message"  
1382 attribute. If the Printer object supports this attribute then it MUST be returned in the response.  
1383 If this attribute is not returned in the response, the client can assume that the "job-state-message"  
1384 attribute is not supported and will not be returned in a subsequent Job object query.

1385

1386 "number-of-intervening-jobs":

1387 The Printer object OPTIONALLY returns the Job object's OPTIONAL "number-of-intervening-  
1388 jobs" attribute. If the Printer object supports this attribute then it MUST be returned in the  
1389 response. If this attribute is not returned in the response, the client can assume that the "number-  
1390 of-intervening-jobs" attribute is not supported and will not be returned in a subsequent Job object  
1391 query.

1392

1393 Note: Since any printer state information which affects a job's state is reflected in the "job-state"  
1394 and "job-state-reasons" attributes, it is sufficient to return only these attributes and no specific  
1395 printer status attributes.

1396

1397 Note: In addition to the MANDATORY common elements required for every operation response, The the  
1398 simplest response consists of the just the "attributes-charset" and "attributes-natural-language" operation  
1399 attributes and the MANDATORY "job-uri", "job-id", and "job-state" Job Object Attributes, the  
1400 MANDATORY "attributes-charset" and "attributes-natural-language" operation attributes, In this



1401 simplest case, the status code is and a status code of "successful-ok" and there is no "status-message"  
1402 operation attribute.

### 1403 3.2.2 Print-URI Operation

1404 This OPTIONAL operation is identical to the Print-Job operation (section 3.2.1) except that a client  
1405 supplies a URI reference to the document data using the "document-uri" (uri) operation attribute (in  
1406 Group 1) rather than including the document data itself. Before returning the response, the Printer  
1407 MUST validate that the Printer supports the retrieval method (e.g., http, ftp, etc.) implied by the URI,  
1408 and MUST check for valid URI syntax. If the client-supplied URI scheme is not supported, i.e. the value  
1409 is not in the Printer object's "referenced-uri-scheme-supported" attribute, the Printer object SHALL reject  
1410 the request and return the 'client-error-uri-scheme-not-supported' status code. See Section 16.3.5 for  
1411 suggested additional checks. The Printer NEED NOT follow the reference and validate the contents of  
1412 the reference.

1413 If the Printer object supports this operation, it MUST support the "reference-uri-schemes-supported"  
1414 Printer attribute (see section 4.4.24).

1415 It is up to the IPP object to interpret the URI and subsequently "pull" the document from the source  
1416 referenced by the URI string.

### 1417 3.2.3 Validate-Job Operation

1418 This MANDATORY operation is similar to the Print-Job operation (section 3.2.1) except that a client  
1419 supplies no document data and the Printer allocates no resources (i.e., it does not create a new Job  
1420 object). This operation is used only to verify capabilities of a printer object against whatever attributes  
1421 are supplied by the client in the Validate-Job request. By using the Validate-Job operation a client can  
1422 check-validate that ~~the same an identical~~ Print-Job operation (with the document data) would be accepted.  
1423 will be accepted without having to send the document data. The Validate-Job operation also performs  
1424 the same security negotiation as the Print-Job operation (see section 8), so that a client can check that the  
1425 client and Printer object security requirements can be met before performing a Print-Job operation.

1426 Note: The Validate-Job operation does not accept a "document-uri" attribute in order to allow a client to  
1427 check that the same Print-URI operation will be accepted, since the client doesn't send the data with the  
1428 Print-URI operation. The client SHOULD just issue the Print-URI request.

1429 The Printer object returns the same status codes, Operation Attributes (Group 1) and Unsupported  
1430 Attributes (Group 2) as the Print-Job operation. However, no Job Object Attributes (Group 3) are  
1431 returned, since no Job object is created.



### 1432 3.2.4 Create-Job Operation

1433 This OPTIONAL operation is similar to the Print-Job operation (section 3.2.1) except that in the Create-  
1434 Job request, a client does not supply document data (or any reference to document data). Also, the client  
1435 does not supply any of the "document-name", "document-format", "compression", or "document-natural-  
1436 language" operation attributes. This operation is followed by one or more Send-Document or Send-URI  
1437 operations. In each of those operation requests, the client OPTIONALLY supplies the "document-  
1438 name", "document-format", and "document-natural-language" attributes for each document in the multi-  
1439 document Job object. If a Printer object supports the Create-Job operation, it MUST also support the  
1440 Send-Document operation and also MAY support the Send-URI operation.

### 1441 3.2.5 Get-Printer-Attributes Operation

1442 This MANDATORY operation allows a client to request the values of the attributes of a Printer object.  
1443 In the request, the client supplies the set of Printer attribute names and/or attribute group names in which  
1444 the requester is interested. In the response, the Printer object returns a corresponding attribute set with  
1445 the appropriate attribute values filled in.

1446 For Printer objects, the possible names of attribute groups are:

- 1447 - 'job-template': all of the Job Template attributes that apply to a Printer object (the last two columns  
1448 of the table in Section 4.2).
- 1449 - 'printer-description': the attributes specified in Section 4.4.
- 1450 - 'all': the special group 'all' that includes all supported attributes.

1451  
1452 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
1453 overlap. For example, if a client requests, 'printer-name' and 'all', the client is actually requesting the  
1454 "printer-name" attribute twice: once by naming it explicitly, and once by inclusion in the 'all' group. In  
1455 such cases, the Printer object NEED NOT return each attribute only once in the response even if it is  
1456 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1457 It is NOT REQUIRED that a Printer object support all attributes belonging to a group (since some  
1458 attributes are OPTIONAL). However, it is MANDATORY that each Printer object support all group  
1459 names.

#### 1460 3.2.5.1 Get-Printer-Attributes Request

1461 The following sets of attributes are part of the Get-Printer-Attributes Request:

#### 1462 Group 1: Operation Attributes

1463 Natural Language and Character Set:

1464 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1465 3.1.4.1.

1466

1467 Target:

1468 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1469 section 3.1.5.

1470

1471 Requesting User Name:

1472 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1473 described in section 8.3.

1474

1475 "requested-attributes" (1setOf keyword) :

1476 The client OPTIONALLY supplies a set of attribute names and/or attribute group names in whose  
1477 values the requester is interested. The Printer object MUST support this attribute. If the client  
1478 omits this attribute, the Printer SHALL respond as if this attribute had been supplied with a value  
1479 of 'all'.

1480

1481 "document-format" (mimeMediaType) :

1482 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1483 attribute. This attribute is useful for a Printer object to determine the set of supported attribute  
1484 values that relate to the requested document format. The Printer object SHALL return the  
1485 attributes and values that it uses to validate a job on a create or Validate-Job operation in which  
1486 this document format is supplied. The Printer object SHOULD return only (1) those attributes  
1487 that are supported for the specified format and (2) the attribute values that are supported for the  
1488 specified document format. By specifying the document format, the client can get the Printer  
1489 object to eliminate the attributes and values that are not supported for a specific document format.  
1490 For example, a Printer object might have multiple interpreters to support both  
1491 'application/postscript' (for PostScript) and 'text/plain' (for text) documents. However, for only  
1492 one of those interpreters might the Printer object be able to support "number-up" with values of  
1493 '1', '2', and '4'. For the other interpreter it might be able to only support "number-up" with a value  
1494 of '1'. ~~If the Printer object does not distinguish between different document formats when  
1495 validating jobs in the create and Validate-Job operations, it SHALL NOT distinguish between  
1496 document formats in the Get-Printer-Attributes operation.~~ Thus a client can use the Get-Printer-  
1497 Attributes operation to obtain the attributes and values that will be used to accept/reject a create  
1498 job operation.

1499

1500 Note: If the Printer object does not distinguish between different sets of supported values for  
1501 each different document format when validating jobs in the create and Validate-Job operations, it  
1502 SHALL NOT distinguish between different document formats in the Get-Printer-Attributes

1503 operation. If the Printer object does distinguish between different sets of supported values for  
1504 each different document format specified by the client, this specialization applies only to the  
1505 following Printer object attributes:

- 1506
- 1507 - Printer attributes that are Job Template attributes ("xxx-default" and xxx"-supported in the
- 1508 Table in Section 4.2),
- 1509 - "pdl-override-supported",
- 1510 - "compression-supported",
- 1511 - "job-k-octets-supported",
- 1512 - "job-impressions-supported,
- 1513 - "job-media-sheets-supported"
- 1514 - "printer-driver-installer",
- 1515 - "color-supported", and
- 1516 - "reference-uri-schemes-supported"

1517

1518 The values of all other Printer object attributes (including "document-format-supported") remain  
1519 invariant with respect to the client supplied document format.

1520

1521 If the client omits this "document-format" operation attribute, the Printer object SHALL respond  
1522 as if the attribute had been supplied with ~~the a value equal to the~~ value of the Printer object's  
1523 "document-format-default" attribute. It is recommended that the client always supply a value for  
1524 "document-format", since the Printer object's "document-format-default" may be  
1525 'application/octet-stream', in which case the returned attributes and values are for the union of the  
1526 document formats that the Printer can automatically sense. For more details, see the description  
1527 of the 'mimeMediaType' attribute syntax in section 4.1.9.

1528

1529 If the client supplies a value for the "document-format" Operation attribute that is not supported  
1530 by the Printer, i.e., is not among the values of the Printer object's "document-format-supported"  
1531 attribute, the Printer object SHALL reject the operation and return the 'client-error-document-  
1532 format-not-supported' status code.

1533

### 1534 3.2.5.2 Get-Printer-Attributes Response

1535 The Printer object returns the following sets of attributes as part of the Get-Printer-Attributes Response:

#### 1536 Group 1: Operation Attributes

1537 Status ~~Code and~~ Message:

1538 In addition to the MANDATORY status code returned in every response, the response  
1539 OPTIONALLY includes a ~~The response includes the MANDATORY status code and an~~  
1540 ~~OPTIONAL~~ "status-message" (text) operation attribute as described in section 3.1.5.

1541

1542 Natural Language and Character Set:

1543 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1544 3.1.4.2.

1545

1546 Group 2: Unsupported Attributes

1547 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1548 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16).

1549

1550 Group 3: Printer Object Attributes

1551 This is the set of requested attributes and their current values. The Printer object ignores (does  
1552 not respond with) any requested attribute which is not supported. The Printer object MAY  
1553 respond with a subset of the supported attributes and values, depending on the security policy in  
1554 force. However, the Printer object SHALL respond with the 'unknown' value for any supported  
1555 attribute (including all MANDATORY attributes) for which the Printer object does not know the  
1556 value. Also the Printer object SHALL respond with the 'no-value' for any supported attribute  
1557 (including all MANDATORY attributes) for which the system administrator has not configured a  
1558 value. See the description of the "out-of-band" values in the beginning of Section 4.1.

1559

1560 3.2.6 Get-Jobs Operation

1561 This MANDATORY operation allows a client to retrieve the list of Job objects belonging to the target  
1562 Printer object. The client may also supply a list of Job attribute names and/or attribute group names. A  
1563 group of Job object attributes will be returned for each Job object that is returned.

1564 This operation is similar to the Get-Job-Attributes operation, except that this Get-Jobs operation returns  
1565 attributes from possibly more than one object (see the description of Job attribute group names in section  
1566 3.3.4).

1567 3.2.6.1 Get-Jobs Request

1568 The client submits the Get-Jobs request to a Printer object.

1569 The following groups of attributes are part of the Get-Jobs Request:

## 1570 Group 1: Operation Attributes

## 1571 Natural Language and Character Set:

1572 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1573 3.1.4.1.

1574

## 1575 Target:

1576 The "printer-uri" (uri) operation attribute which is the target for this operation as described in  
1577 section 3.1.5.

1578

## 1579 Requesting User Name:

1580 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1581 described in section 8.3.

1582

## 1583 "limit" (integer(1:MAX)):

1584 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1585 attribute. It is an integer value that indicates a limit to the number of Job objects returned. The  
1586 limit is a "stateless limit" in that if the value supplied by the client is 'N', then only the first 'N' jobs  
1587 are returned in the Get-Jobs Response. There is no mechanism to allow for the next 'M' jobs after  
1588 the first 'N' jobs. If the client does not supply this attribute, the Printer object responds with all  
1589 applicable jobs.

1590

## 1591 "requested-attributes" (1setOf keyword):

1592 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1593 attribute. It is a set of Job attribute names and/or attribute groups names in whose values the  
1594 requester is interested. This set of attributes is returned for each Job object that is returned. The  
1595 allowed attribute group names are the same as those defined in the Get-Job-Attributes operation  
1596 in section 3.3.4. If the client does not supply this attribute, the Printer SHALL respond as if the  
1597 client had supplied this attribute with two values: 'job-uri' and 'job-id'.

1598

## 1599 "which-jobs" (keyword):

1600 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1601 attribute. It indicates which Job objects SHALL be returned by the Printer object. The values for  
1602 this attribute are:

1603

1604 'completed': This includes any Job object whose state is 'completed', 'canceled', or 'aborted'.

1605 'not-completed': This includes any Job object whose state is 'pending', 'processing',

1606 'processing-stopped', or 'pending-held'.

1607

1608 A Printer object SHALL support both values. However, if the implementation does not keep jobs  
1609 in the 'completed', 'canceled', and 'aborted' states, then it returns no jobs when the 'completed'  
1610 value is supplied.

1611  
1612 If a client supplies some other value, the Printer object SHALL copy the attribute and the  
1613 unsupported value to the Unsupported Attributes response group, reject the request, and return  
1614 the 'client-error-attributes-or-values-not-supported' status code.

1615  
1616 If the client does not supply this attribute, the Printer object SHALL respond as if the client had  
1617 supplied the attribute with a value of 'not-completed'.

1618  
1619 "my-jobs" (boolean):  
1620 The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1621 attribute. It indicates whether all jobs or just the jobs submitted by the requesting user of this  
1622 request SHALL be returned by the Printer object. If the client does not supply this attribute, the  
1623 Printer object SHALL respond as if the client had supplied the attribute with a value of 'false', i.e.,  
1624 all jobs. The means for authenticating the requesting user and matching the jobs is described in  
1625 section 8.

### 1626 3.2.6.2 Get-Jobs Response

1627 The Printer object returns all of the Job objects that match the criteria as defined by the attribute values  
1628 supplied by the client in the request. It is possible that no Job objects are returned since there may  
1629 literally be no Job objects at the Printer, or there may be no Job objects that match the criteria supplied by  
1630 the client. If the client requests any Job attributes at all, there is a set of Job Object Attributes returned  
1631 for each Job object.

#### 1632 Group 1: Operation Attributes

1633 Status ~~Code and~~ Message:

1634 In addition to the MANDATORY status code returned in every response, the response  
1635 OPTIONALLY includes a ~~The response includes the MANDATORY status code and an~~  
1636 ~~OPTIONAL~~ "status-message" (text) operation attribute as described in section 3.1.5.

1637  
1638 Natural Language and Character Set:

1639 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1640 3.1.4.2.

1641  
1642 Group 2: Unsupported Attributes

1643 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1644 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).  
1645

### 1646 Groups 3 to N: Job Object Attributes

1647 The Printer object responds with one set of Job Object Attributes for each returned Job object.  
1648 The Printer object ignores (does not respond with) any requested attribute or value which is not  
1649 supported or which is restricted by the security policy in force, including whether the requesting  
1650 user is the user that submitted the job (job originating user) or not (see section 8). However, the  
1651 Printer object SHALL respond with the 'unknown' value for any supported attribute (including all  
1652 MANDATORY attributes) for which the Printer object does not know the value, unless it would  
1653 violate the security policy. See the description of the "out-of-band" values in the beginning of  
1654 Section 4.1.

1655  
1656 For any job submitted in a different natural language than the natural language that the Printer  
1657 object is returning in the "attributes-natural-language" operation attribute in the Get-Jobs  
1658 response, the Printer SHALL indicate the submitted natural language by returning the Job object's  
1659 "attributes-natural-language" as the first Job object attribute, which overrides the "attributes-  
1660 natural-language" operation attribute value being returned by the Printer object. If any returned  
1661 'text' or 'name' attribute includes a Natural Language Override as described in the sections 4.1.1.2  
1662 and 4.1.2.2, the Natural Language Override overrides the Job object's "attributes-natural-  
1663 language" value and/or the "attributes-natural-language" operation attribute value.  
1664

1665 Jobs are returned in the following order:

- 1666 - If the client requests all 'completed' Jobs (Jobs in the 'completed', 'aborted', or 'canceled'  
1667 states), then the Jobs are returned newest to oldest (with respect to actual completion  
1668 time)
- 1669 - If the client requests all 'not-completed' Jobs (Jobs in the 'pending', 'processing', 'pending-  
1670 held', and 'processing-stopped' states), then Jobs are returned in relative chronological  
1671 order of expected time to complete (based on whatever scheduling algorithm is configured  
1672 for the Printer object).  
1673

### 1674 3.3 Job Operations

1675 All Job operations are directed at Job objects. A client MUST always supply some means of identifying  
1676 the Job object in order to identify the correct target of the operation. That job identification MAY either  
1677 be a single Job URI or a combination of a Printer URI with a Job ID. The IPP object implementation  
1678 MUST support both forms of identification for every job.

## 1679 3.3.1 Send-Document Operation

1680 This OPTIONAL operation allows a client to create a multi-document Job object that is initially "empty"  
1681 (contains no documents). In the Create-Job response, the Printer object returns the Job object's URI (the  
1682 "job-uri" attribute) and the Job object's 32-bit identifier (the "job-id" attribute). For each new document  
1683 that the client desires to add, the client uses a Send-Document operation. Each Send-Document Request  
1684 contains the entire stream of document data for one document.

1685 Since the Create-Job and the send operations (Send-Document or Send-URI operations) that follow can  
1686 occur over arbitrarily long periods of time, each Printer object must decide how long to "wait" for the  
1687 next send operation. The Printer object OPTIONALLY supports the "multiple-operation-timeout"  
1688 attribute. This attribute indicates the maximum number of seconds the Printer object will wait for the  
1689 next send operation. If the Printer object times-out waiting for the next send operation, the Printer object  
1690 MAY decide on any of the following semantic actions:

- 1691 1. Assume that the Job is an invalid job, start the process of changing the job state to 'aborted', and  
1692 clean up all resources associated with the Job. In this case, if another send operation is finally  
1693 received, the Printer responds with an "client-error-not-possible" or "client-error-not-found"  
1694 depending on whether or not the Job object is still around when it finally arrives.
- 1695 2. Assume that the last send operation received was in fact the last document (as if the "last-  
1696 document" flag had been set to 'true'), close the Job object, and proceed to process it (i.e., move  
1697 the Job's state to 'pending').
- 1698 3. Assume that the last send operation received was in fact the last document, close the Job, but move  
1699 it to the 'pending-held' to allow an operator to determine whether or not to continue processing  
1700 the Job by moving it back to the 'pending' state.

1701  
1702 Each implementation is free to decide the "best" action to take depending on local policy, the value of  
1703 "ipp-attribute-fidelity", and/or any other piece of information available to it. If the choice is to abort the  
1704 Job object, it is possible that the Job object may already have been processed to the point that some  
1705 media sheet pages have been printed.

## 1706 3.3.1.1 Send-Document Request

1707 The following attribute sets are part of the Send-Document Request:

## 1708 Group 1: Operation Attributes

## 1709 Natural Language and Character Set:

1710 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1711 3.1.4.1.

1712



1713 Target:  
1714     Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1715     attribute(s) which define the target for this operation as described in section 3.1.5.  
1716

1717 Requesting User Name:  
1718     The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1719     described in section 8.3.  
1720

1721 "document-name" (name(MAX)):  
1722     The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1723     attribute. It contains the client supplied document name. The document name MAY be different  
1724     than the Job name. It might be helpful, but NEED NOT be unique across multiple documents in  
1725     the same Job. Typically, the client software automatically supplies the document name on behalf  
1726     of the end user by using a file name or an application generated name. See the description of the  
1727     "document-name" operation attribute in the Print-Job Request (section 3.2.1.1) for more  
1728     information about this attribute.  
1729

1730 "document-format" (mimeMediaType) :  
1731     The client OPTIONALLY supplies this attribute. The Printer object MUST support this  
1732     attribute. The value of this attribute identifies the format of the supplied document data. If the  
1733     client does not supply this attribute, the Printer object assumes that the document data is in the  
1734     format defined by the Printer object's "document-format-default" attribute. If the client supplies  
1735     this attribute, but the value is not supported by the Printer object, i.e., the value is not one of the  
1736     values of the Printer object's "document-format-supported" attribute, the Printer object SHALL  
1737     reject the request and return the 'client-error-document-format-not-supported' status code.  
1738

1739 "document-natural-language" (naturalLanguage):  
1740     The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1741     this attribute. This attribute specifies the natural language of the document for those document-  
1742     formats that require a specification of the natural language in order to image the document  
1743     unambiguously. There are no particular values required for the Printer object to support.  
1744

1745 "compression" (type3 keyword)  
1746     The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1747     this attribute and the "compression-supported" attribute (see section 4.4.29). The client supplied  
1748     "compression" operation attribute identifies the compression algorithm used on the document  
1749     data. If the client omits this attribute, the Printer object SHALL assume that the data is not  
1750     compressed. If the client supplies the attribute and the Printer object supports the attribute, the  
1751     Printer object SHALL use the corresponding decompression algorithm on the document data. If  
1752     the client supplies this attribute, but the value is not supported by the Printer object, i.e., the value

1753 is not one of the values of the Printer object's "compression-supported" attribute, the Printer  
1754 object SHALL copy the attribute and its value to the Unsupported Attributes response group,  
1755 reject the request, and return the 'client-error-attributes-or-values-not-supported' status code.  
1756

1757 "last-document" (boolean):

1758 The client MUST supply this attribute. The Printer object MUST support this attribute. It is a  
1759 boolean flag that is set to 'true' if this is the last document for the Job, 'false' otherwise.  
1760

## 1761 Group 2: Document Content

1762 The client MUST supply the document data if the "last-document" flag is set to 'false'. However,  
1763 since a client might not know that the previous document sent with a Send-Document (or Send-  
1764 URI) operation was the last document (i.e., the "last-document" attribute was set to 'false'), it is  
1765 legal to send a Send-Document request with no document data where the "last-document" flag is  
1766 set to 'true'. Such a request SHALL NOT increment the value of the Job object's "number-of-  
1767 documents" attribute, since no real document was added to the job.

### 1768 3.3.1.2 Send-Document Response

1769 The following sets of attributes are part of the Send-Document Response:

#### 1770 Group 1: Operation Attributes

1771 Status ~~Code and~~ Message:

1772 In addition to the MANDATORY status code returned in every response, the response  
1773 OPTIONALLY includes a ~~The response includes the MANDATORY status code and an~~  
1774 ~~OPTIONAL~~- "status-message" (text) operation attribute as described in section 3.1.5.  
1775

1776 Natural Language and Character Set:

1777 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1778 3.1.4.2.  
1779

#### 1780 Group 2: Unsupported Attributes

1781 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1782 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).  
1783

#### 1784 Group 3: Job Object Attributes

1785 This is the same set of attributes as described in the Print-Job response (see section 3.2.1.2).  
1786

## 1787 3.3.2 Send-URI Operation

1788 This OPTIONAL operation is identical to the Send-Document operation (see section 3.3.1) except that a  
1789 client MUST supply a URI reference ("document-uri" operation attribute) rather than the document data  
1790 itself. If a Printer object supports this operation, clients can use both Send-URI or Send-Document  
1791 operations to add new documents to an existing multi-document Job object. However, if a client needs  
1792 to indicate that the previous Send-URI or Send-Document was the last document, the client MUST use  
1793 the Send-Document operation with no document data and the "last-document" flag set to 'true' (rather  
1794 than using a Send-URI operation with no "document-uri" operation attribute). If a Printer object  
1795 supports this operation, it MUST also support the Print-URI operation (see section 3.2.2).

1796 The Printer object MUST validate the syntax and URI scheme of the supplied URI before returning a  
1797 response, just as in the Print-URI operation.

## 1798 3.3.3 Cancel-Job Operation

1799 This MANDATORY operation allows a client to cancel a Print Job any time after a create job operation.  
1800 Since a Job might already be printing by the time a Cancel-Job is received, some media sheet pages might  
1801 be printed before the job is actually terminated.

## 1802 3.3.3.1 Cancel-Job Request

1803 The following groups of attributes are part of the Cancel-Job Request:

## 1804 Group 1: Operation Attributes

1805 Natural Language and Character Set:

1806 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1807 3.1.4.1.

1808  
1809 Target:

1810 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1811 attribute(s) which define the target for this operation as described in section 3.1.5.

1812  
1813 Requesting User Name:

1814 The "requesting-user-name" ([name\(MAX\)](#)) attribute SHOULD be supplied by the client as  
1815 described in section 8.3.

1816  
1817 "message" (text(127)):

1818 The client OPTIONALLY supplies this attribute. The Printer object OPTIONALLY supports  
1819 this attribute. It is a message to the operator. This "message" attribute is not the same as the "job-

1820 message-from-operator" attribute. That attribute is used to report a message from the operator to  
1821 the end user that queries that attribute. This "message" operation attribute is used to send a  
1822 message from the client to the operator along with the operation request. It is an implementation  
1823 decision of how or where to display this message to the operator (if at all).  
1824

### 1825 3.3.3.2 Cancel-Job Response

1826 The following sets of attributes are part of the Cancel-Job Response:

#### 1827 Group 1: Operation Attributes

1828 Status ~~Code and~~ Message:

1829 In addition to the MANDATORY status code returned in every response, the response  
1830 OPTIONALLY includes a ~~The response includes the MANDATORY status code and an~~  
1831 ~~OPTIONAL~~ "status-message" (text) operation attribute as described in section 3.1.5.  
1832

1833 If the job is already in the 'completed', 'aborted', or 'canceled' state, or the 'process-to-stop-point'  
1834 value is set in the Job's "job-state-reasons" attribute, the Printer object SHALL reject the request  
1835 and return the 'client-error-not-possible' error status code.  
1836

1837 Natural Language and Character Set:

1838 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1839 3.1.4.2.  
1840

#### 1841 Group 2: Unsupported Attributes

1842 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1843 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).  
1844

1845 Once a successful response has been sent, the implementation guarantees that the Job will eventually end  
1846 up in the 'canceled' state. Between the time of the Cancel-Job operation is accepted and when the job  
1847 enters the 'canceled' job-state (see section 4.3.7), the "job-state-reasons" attribute SHOULD contain the '  
1848 processing-to-stop-point' value which indicates to later queries that although the Job might still be  
1849 'processing', it will eventually end up in the 'canceled' state, not the 'completed' state.

### 1850 3.3.4 Get-Job-Attributes Operation

1851 This MANDATORY operation allows a client to request the values of attributes of a Job object and it is  
1852 almost identical to the Get-Printer-Attributes operation (see section 3.2.5). The only differences are that

1853 the operation is directed at a Job object rather than a Printer object, there is no "document-format"  
1854 operation attribute used when querying a Job object, and the returned attribute group is a set of Job  
1855 object attributes rather than a set of Printer object attributes.

1856 For Jobs, the possible names of attribute groups are:

- 1857 - 'job-template': all of the Job Template attributes that apply to a Job object (the first column of the  
1858 table in Section 4.2).
- 1859 - 'job-description': all of the Job Description attributes specified in Section 4.3.
- 1860 - 'all': the special group 'all' that includes all supported attributes.

1861

1862 Since a client MAY request specific attributes or named groups, there is a potential that there is some  
1863 overlap. For example, if a client requests, 'job-name' and 'job-description', the client is actually requesting  
1864 the "job-name" attribute once by naming it explicitly, and once by inclusion in the 'job-description' group.  
1865 In such cases, the Printer object NEED NOT return the attribute only once in the response even if it is  
1866 requested multiple times. The client SHOULD NOT request the same attribute in multiple ways.

1867 It is NOT REQUIRED that a Job object support all attributes belonging to a group (since some attributes  
1868 are OPTIONAL). However it is MANDATORY that each Job object support all group names.

#### 1869 3.3.4.1 Get-Job-Attributes Request

1870 The following groups of attributes are part of the Get-Job-Attributes Request when the request is  
1871 directed at a Job object:

##### 1872 Group 1: Operation Attributes

1873 Natural Language and Character Set:

1874 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1875 3.1.4.1.

1876

1877 Target:

1878 Either (1) the "printer-uri" (uri) plus "job-id" (integer(1:MAX)) or (2) the "job-uri" (uri) operation  
1879 attribute(s) which define the target for this operation as described in section 3.1.5.

1880

1881 Requesting User Name:

1882 The "requesting-user-name" (name(MAX)) attribute SHOULD be supplied by the client as  
1883 described in section 8.3.

1884

1885 "requested-attributes" (1setOf keyword) :

1886 The client OPTIONALLY supplies this attribute. The IPP object MUST support this attribute.  
1887 It is a set of attribute names and/or attribute group names in whose values the requester is  
1888 interested. If the client omits this attribute, the IPP object SHALL respond as if this attribute had  
1889 been supplied with a value of 'all'.  
1890

#### 1891 3.3.4.2 Get-Job-Attributes Response

1892 The Printer object returns the following sets of attributes as part of the Get-Job-Attributes Response:

##### 1893 Group 1: Operation Attributes

1894 Status ~~Code and~~ Message:

1895 In addition to the MANDATORY status code returned in every response, the response  
1896 OPTIONALLY includes a ~~The response includes the MANDATORY status code and an~~  
1897 ~~OPTIONAL~~ "status-message" (text) operation attribute as described in section 3.1.5.  
1898

1899 Natural Language and Character Set:

1900 The "attributes-charset" and "attributes-natural-language" attributes as described in section  
1901 3.1.4.2. The "attributes-natural-language" MAY be the natural language of the Job object, rather  
1902 than the one requested.  
1903

##### 1904 Group 2: Unsupported Attributes

1905 This is a set of Operation attributes supplied by the client (in the request) that are not supported  
1906 by the Printer object or that conflict with one another (see sections 3.2.1.2 and 16.3).  
1907

##### 1908 Group 3: Job Object Attributes

1909 This is the set of requested attributes and their current values. The IPP object ignores (does not  
1910 respond with) any requested attribute or value which is not supported or which is restricted by the  
1911 security policy in force, including whether the requesting user is the user that submitted the job  
1912 (job originating user) or not (see section 8). However, the IPP object SHALL respond with the  
1913 'unknown' value for any supported attribute (including all MANDATORY attributes) for which  
1914 the IPP object does not know the value, unless it would violate the security policy. See the  
1915 description of the "out-of-band" values in the beginning of Section 4.1.

## 1916 4. Object Attributes

1917 This section describes the attributes with their corresponding attribute syntaxes and values that are part of  
1918 the IPP model. The sections below show the objects and their associated attributes which are included  
1919 within the scope of this protocol. Many of these attributes are derived from other relevant specifications:

1920 - Document Printing Application (DPA) [ISO10175]

1921 - RFC 1759 Printer MIB [RFC1759]

1922

1923 Each attribute is uniquely identified in this document using a "keyword" (see section 13.2.1) which is the  
1924 name of the attribute. The keyword is included in the section header describing that attribute.

1925 Note: Not only are keywords used to identify attributes, but one of the attribute syntaxes described  
1926 below is "keyword" so that some attributes have keyword values. Therefore, these attributes are defined  
1927 as having an attribute syntax that is a set of keywords.

## 1928 4.1 Attribute Syntaxes

1929 This section defines the basic attribute syntax types that all clients and IPP objects SHALL be able to  
1930 accept in responses and accept in requests, respectively. Each attribute description in sections 3 and 4  
1931 includes the name of attribute syntax(es) in the heading (in parentheses). A conforming implementation  
1932 of an attribute SHALL include the semantics of the attribute syntax(es) so identified. Section **Error!**  
1933 **Reference source not found.** describes how the protocol can be extended with new attribute syntaxes.

1934 The attribute syntaxes are specified in the following sub-sections, where the sub-section heading is the  
1935 keyword name of the attribute syntax inside the single quotes. In operation requests and responses each  
1936 attribute value MUST be represented as one of the attribute syntaxes specified in the sub-section heading  
1937 for the attribute. In addition, the value of an attribute in a response (but not in a request) MAY be one of  
1938 the "out-of-band" values. Standard "out-of-band" values are:

1939 'unknown': The attribute is supported by the IPP object, but the value is unknown to the IPP object  
1940 for some reason.

1941 'unsupported': The attribute is unsupported by the IPP object. This value SHALL be returned only as  
1942 the value of an attribute in the Unsupported Attributes Group.

1943 'no-value': The attribute is supported by the Printer object, but the system administrator has not yet  
1944 configured a value.

1945

1946 The protocol specification defines mechanisms for passing "out-of-band" values. All attributes in a  
1947 request SHALL have one or more values as defined in Sections 4.2 to 4.4. Thus clients SHALL ~~not~~ |



1948 NOT supply attributes with "out-of-band" values. All attribute in a response SHALL have one or more  
1949 values as defined in Sections 4.2 to 4.4 or a single "out-of-band" value.

1950 Most attributes are defined to have a single attribute syntax. However, a few attributes (e.g., "job-sheet",  
1951 "media", "job-hold-until") are defined to have several attribute syntaxes, depending on the value. These  
1952 multiple attribute syntaxes are separated by the "|" character in the sub-section heading to indicate the  
1953 choice. Since each value SHALL be tagged as to its attribute syntax in the protocol, a single-valued  
1954 attribute instance may have any one of its attribute syntaxes and a multi-valued attribute instance may  
1955 have a mixture of its defined attribute syntaxes.

#### 1956 4.1.1 'text'

1957 A text attribute is an attribute whose value is a sequence of one or more  
1958 characters encoded in a maximum of 1023 ('MAX') octets. MAX is the maximum length for all values of  
1959 any text attribute. However, if an attribute will always contain values whose maximum length is much  
1960 less than MAX, the definition of that attribute will include a qualifier that defines the maximum length for  
1961 values of that attribute. For example: the "printer-location" attribute is specified as "printer-location  
1962 (text(127))". In this case, text values for "printer-location" SHALL NOT exceed 127 octets; if supplied  
1963 with a longer text string via some external interface, implementations are free to truncate to this shorter  
1964 length limitation.

1965 In this specification, all text attributes are defined using the 'text' syntax. However, 'text' is used only for  
1966 brevity; the formal interpretation of 'text' is: 'textWithoutLanguage | textWithLanguage'. That is, for any  
1967 attribute defined in this specification using the 'text' attribute syntax, all IPP objects and clients SHALL  
1968 accept, support, and return either the 'textWithoutLanguage' or 'textWithLanguage' attribute syntaxes in  
1969 actual usage and protocol execution. The syntax 'text' never appears "on-the-wire".

1970 Both 'textWithoutLanguage' and 'textWithLanguage' are needed to support the real world needs of  
1971 interoperability between sties and systems that use different natural languages as the basis for human  
1972 communication. Generally, one natural language applies to all text attributes in a give request or  
1973 response. The language is indicated by the "attributes-natural-language" operation attribute defined in  
1974 section 3.1.4 or "attributes-natural-language" job attribute defined in section 4.3.24, and there is no need  
1975 to identify the natural language for each text string on a value-by-value basis. In these cases, the attribute  
1976 syntax 'textWithoutLanguage' is used for text attributes. In other cases, the client needs to supply or the  
1977 Printer object needs to return a text value in a natural language that is different from the rest of the text  
1978 values in the request or response. In these cases, the client or Printer object uses the attribute syntax  
1979 'textWithLanguage' for text attributes (this is the Natural Language Override mechanism described in  
1980 section 3.1.4).

1981 'textWithoutLanguage' and 'textWithLanguage' are described in more detail in the following sections.



1982 4.1.1.1 'textWithoutLanguage'

1983 ~~The 'textWithoutLanguage' syntax indicates a value that is sequence of zero or more characters. which is~~  
1984 ~~indicated in sub-section headers using the notation: text(MAX). If an attribute is specified to have a~~  
1985 ~~smaller maximum in its sub-section header description, the explicit number of octets is indicated. For~~  
1986 ~~example: the "printer-location" attribute is specified as: printer-location (text(127)). Text strings are~~  
1987 ~~encoded using the rules of some charset.~~ The Printer object SHALL support the UTF-8 charset  
1988 [RFC2044] and MAY support additional charsets to represent 'text' values, provided that the charsets are  
1989 registered with IANA [IANA-CS]. See Section 4.1.7 for the specification of the 'charset' attribute  
1990 syntax, including restricted semantics and examples of charsets.

1991 ~~In this specification, attributes that are indicated to have the 'text' attribute syntax, also automatically have~~  
1992 ~~the 'textWithLanguage' attribute syntax. See section 4.1.2.~~

1993 ~~If the client needs to supply or the Printer object needs to return a text value in a different natural~~  
1994 ~~language from the rest of the 'text' attributes in the request or response as indicated by the "attributes-~~  
1995 ~~natural-language" operation attribute (see Section 3.1.3) or job attribute (see Section 4.3.24), the client~~  
1996 ~~or Printer object SHALL identify the natural language for that attribute. This MANDATORY~~  
1997 ~~mechanism for identifying the natural language of a single attribute value is called the Natural Language~~  
1998 ~~Override mechanism. This mechanism uses an alternate attribute syntax, called 'textWithLanguage',~~  
1999 ~~which is described in section 4.1.2.~~

2000 4.1.24.1.1.2 'textWithLanguage'

2001 The 'textWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2002 'textWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides the  
2003 natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that applies  
2004 to the text part of that value and that value alone. ~~For any give text attribute, The-the~~  
2005 ~~'textWithoutLanguage' part is limited to the maximum length defined for that attribute, 1023 octets, and~~  
2006 ~~but the 'naturalLanguage' part is always limited to 63 octets. If the sub-section header specifying an~~  
2007 ~~attribute with attribute syntax 'text' with a smaller explicit value than MAX, that value applies to the 'text'~~  
2008 ~~part of the 'textWithLanguage' as well. Using the 'textWithLanguage' attribute syntax rather than the~~  
2009 ~~normal 'textWithoutLanguage' syntax is the so-called Natural Language Override mechanism and MUST~~  
2010 ~~be supported by all IPP objects and clients.~~

2011 ~~If the client needs to supply or the Printer object needs to return a 'text' attribute value in a different~~  
2012 ~~natural language from the rest of the 'text' attribute values in the request or response as indicated by the~~  
2013 ~~"attributes-natural-language" operation attribute (see Section 3.1.3) or to the "attributes-natural-~~  
2014 ~~language" Job attribute, if present, in the case of a Get Jobs response, the client or IPP object SHALL~~  
2015 ~~identify the natural language for that attribute using the 'textWithLanguage' attribute syntax.~~

2016 ~~The 'textWithLanguage' attribute syntax is the so-called Natural Language Override mechanism for the~~  
2017 ~~'text' attribute syntax and MUST be supported by IPP objects.~~

2018 If the attribute is multi-valued (1setOf text), then the 'textWithLanguage' attribute syntax MUST be used  
2019 to explicitly specify each attribute value whose natural language needs to be overridden. Other values in  
2020 a multi-valued 'text' attribute in a request or a response revert to the natural language of the operation  
2021 attribute or to the "attributes-natural-language" Job attribute, if present, in the case of a Get-Jobs  
2022 response.

2023 ~~Any attribute that is specified to have the 'text' attribute syntax in this document, automatically also has~~  
2024 ~~the 'textWithLanguage' attribute syntax. IPP objects SHALL accept, support, and return both the 'text'~~  
2025 ~~and 'textWithLanguage' attribute syntaxes for any attribute in this specification that is indicated to have~~  
2026 ~~the 'text' attribute syntax. For brevity in this specification, only the 'text' attribute syntax is indicated for~~  
2027 ~~attributes. However, the interpretation of 'text' SHALL be as if it were: 'text | textWithLanguage'.~~

2028 In a create request, the Printer object MUST accept and store with the Job object any natural language in  
2029 the "attributes-natural-language" operation attribute, whether the Printer object supports that natural  
2030 language or not. Furthermore, the Printer object MUST accept and store any 'textWithLanguage'  
2031 attribute value, whether the Printer object supports that natural language or not. These requirements are  
2032 independent of the value of the "ipp-attribute-fidelity" operation attribute that the client MAY supply.

2033 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2034 indicating English, but the value of the "job-name" attribute is in French, the client MUST use the  
2035 'textWithLanguage' attribute syntax with the following two values:

2036 'fr': Natural Language Override indicating French

2037 'Rapport Mensuel': the job name in French

2038

2039 See the Protocol document [IPP-PRO] for a detailed example of the 'textWithLanguage' attribute syntax.

2040 4.1.34.1.2 'name'

2041 ~~This syntax type is used for user-friendly strings, such as a Printer name, that, for humans, are more~~  
2042 ~~meaningful than identifiers. Names are usually never translated from one natural language to another.~~  
2043 The 'name' attribute syntax is essentially the same as 'text', including the MANDATORY support of UTF-  
2044 8 ~~and the Natural Language Override mechanism,~~ except that the sequence of characters is limited so that  
2045 its encoded form ~~is of length 1 to~~ SHALL NOT exceed 255 octets.

2046 ~~-which is indicated in sub-section headers using the notation: name(MAX). If an attribute is specified~~  
2047 ~~to have a smaller maximum, the explicit number of octets is indicated. For example: the "printer-~~

2048 ~~name" attribute is specified as: printer-name (name(127)). This syntax type is used for user-~~  
2049 ~~friendly strings, such as a Printer name, that, for humans, are more meaningful than identifiers.~~  
2050 ~~Also like 'text', 'name' is really an abbreviated notation for either 'nameWithoutLanguage' or~~  
2051 ~~'nameWithLanguage'; all IPP objects and clients MUST support the notion of 'name' attributes using~~  
2052 ~~either the 'nameWithoutLanguage' or the 'nameWithLanguage' syntax during protocol execution. In this~~  
2053 ~~specification, attributes that are indicated to have the 'name' attribute syntax, also automatically have the~~  
2054 ~~'nameWithLanguage' attribute syntax. See section 4.1.4.~~

2055 Note: Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override mechanism.

2056 Some attributes are defined as 'type3 keyword | name'. These attributes support values that are either  
2057 type3 keywords or names. This dual-syntax mechanism enables a site administrator to extend these  
2058 attributes to legally include values that are locally defined by the site administrator. Such names are not  
2059 registered with IANA.

2060 4.1.2.1 Note: Only the 'text' and 'name' attribute syntaxes permit the Natural Language Override  
2061 mechanism. 'nameWithoutLanguage'

2062 The 'nameWithoutLanguage' syntax indicates a value that is sequence of zero or more characters so that  
2063 its encoded form does not exceed 127 octets.

2064 4.1.44.1.2.2 'nameWithLanguage'

2065 The 'nameWithLanguage' attribute syntax behaves that same as is the same as the 'textWithLanguage'  
2066 syntax. If a name is in a language that is different than the rest of the object or operation, then this  
2067 'nameWithLanguage' syntax is used rather than the generic 'nameWithoutLanguage' syntax., including the  
2068 MANDATORY support of UTF-8, except that the length of the 'name' part SHALL not exceed 255  
2069 octets. This attribute syntax is the so-called Natural Language Override mechanism for the 'name'  
2070 attribute syntax and MUST be supported by IPP objects.

2071 The 'nameWithLanguage' attribute syntax is a compound attribute syntax consisting of two parts: a  
2072 'nameWithoutLanguage' part plus an additional 'naturalLanguage' (see section 4.1.8) part that overrides  
2073 the natural language in force. The 'naturalLanguage' part explicitly identifies the natural language that  
2074 applies to the that name value and that name value alone.

2075 Example: If the client supplies the "attributes-natural-language" operation attribute with the value: 'en'  
2076 indicating English, but the "printer-name" attribute is in German, the client MUST use the  
2077 'nameWithLanguage' attribute syntax as follows:

2078 'de': Natural Language Override indicating German  
2079 'Farbdrucker': the Printer name in German

2080

2081 [4.1.54.1.3](#) 'keyword'

2082 The 'keyword' attribute syntax is a sequence of characters, length: 1 to 255, containing only the US-  
2083 ASCII [ASCII] encoded values for lowercase letters ("a" - "z"), digits ("0" - "9"), hyphen ("-"), dot ("."),  
2084 and underscore ("\_"). The first character MUST be a lowercase letter. Furthermore, keywords SHALL  
2085 be in U.S. English.

2086 This syntax type is used for enumerating semantic identifiers of entities in the abstract protocol, i.e.,  
2087 entities identified in this document. Keywords are used as attribute names or values of attributes. Unlike  
2088 'text' and 'name' attribute values, 'keyword' values SHALL NOT use the Natural Language Override  
2089 mechanism, since they SHALL always be US-ASCII and U.S. English.

2090 Keywords are for use in the protocol. A user interface will likely provide a mapping between protocol  
2091 keywords and displayable user-friendly words and phrases which are localized to the natural language of  
2092 the user. While the keywords specified in this document MAY be displayed to users whose natural  
2093 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2094 the user interface is outside the scope of this document.

2095 In the definition for each attribute of this syntax type, the full set of defined keyword values for that  
2096 attribute are listed.

2097 When a keyword is used to represent an attribute (its name), it MUST be unique within the full scope of  
2098 all IPP objects and attributes. When a keyword is used to represent a value of an attribute, it MUST be  
2099 unique just within the scope of that attribute. That is, the same keyword SHALL ~~not~~ NOT be used for  
2100 two different values within the same attribute to mean two different semantic ideas. However, the same  
2101 keyword MAY be used across two or more attributes, representing different semantic ideas for each  
2102 attribute. Section **Error! Reference source not found.** describes how the protocol can be extended with  
2103 new keyword values. Examples of attribute name keywords:

2104 "job-name"

2105 "attributes-charset"

2106

2107 Note: This document uses "type1", "type2", and "type3" prefixes to the "keyword" and "enum" basic  
2108 syntaxes. This extra information applies only to how the set of values defined for attributes with these  
2109 syntaxes can be extended; this extra information is not carried in the protocol itself. "type1" indicates  
2110 that new versions of the IPP standards documents must be revised and issued in order for new values to  
2111 be added. "type2" indicates that IPP Subject Matter Experts must work with IANA to review and  
2112 approve any proposed new values before the new values can be registered. "type3" indicates that IPP  
2113 Subject Matter Experts are not required to review and approve any proposed new values before the new

2114 values can be registered with IANA. These extensibility mechanisms and restrictions are fully described  
2115 in section **Error! Reference source not found.**

#### 2116 4.1.64.1.4 'enum'

2117 The 'enum' attribute syntax is an enumerated integer value that is in the range from ~~2\*\*31-(MIN)~~1 to  
2118 2\*\*31 - 1 (MAX). Each value has an associated 'keyword' name. In the definition for each attribute of  
2119 this syntax type, the full set of possible values for that attribute are listed. This syntax type is used for  
2120 attributes for which there are enum values assigned by other standards, such as SNMP MIBs. A number  
2121 of attribute enum values in this specification are also used for corresponding attributes in other standards  
2122 [RFC1759]. This syntax type is not used for attributes to which the system administrator may assign  
2123 values. Section **Error! Reference source not found.** describes how the protocol can be extended with  
2124 new enum values.

2125 Enum values are for use in the protocol. A user interface will provide a mapping between protocol enum  
2126 values and displayable user-friendly words and phrases which are localized to the natural language of the  
2127 user. While the enum symbols specified in this document MAY be displayed to users whose natural  
2128 language is U.S. English, they MAY be mapped to other U.S. English words for U.S. English users, since  
2129 the user interface is outside the scope of this document.

2130 Note: SNMP MIBs use '2' for 'unknown' which corresponds to the IPP "out-of-band" value 'unknown'.  
2131 See the description of the "out-of-band" values at the beginning of Section 4.1. Therefore, attributes of  
2132 type 'enum' start at '3'.

#### 2133 4.1.74.1.5 'uri'

2134 The 'uri' attribute syntax is any valid Uniform Resource Identifier or URI [RFC1630]. Most often, URIs  
2135 are simply Uniform Resource Locators or URLs [RFC1738] [RFC1808]. The maximum length of URIs  
2136 used within IPP is 1023 octets. Although most other IPP syntax types allow for only lower-cased values,  
2137 this syntax type allows for mixed-case values. The URI and URL standards allow for mixed-case values  
2138 that are case-sensitive.

2139 ~~Note: The mapping of IPP/1.0 onto HTTP/1.1 [IPP-PRO] defines that the URL for the IPP object is~~  
2140 ~~included both inside the operation as an Operation Attribute and outside the operation within the HTTP~~  
2141 ~~layer. The potential exists that these two values reference the same IPP object, but are not literally~~  
2142 ~~identical since one can be a relative URL and the other can be an absolute URL. HTTP/1.1 allows clients~~  
2143 ~~to generate and send a relative URL rather than an absolute URL. A relative URL identifies a resource~~  
2144 ~~with the scope of the HTTP server, but does not include scheme, host or port. The following statements~~  
2145 ~~characterize how URLs should be used in the mapping of IPP onto HTTP/1.1:~~

- 2146 ~~1. Although potentially redundant, a client MUST supply the target of the operation both as an~~  
2147 ~~Operation Attribute (see Section 3.1.4) and as a URL at the HTTP layer. The rationale for this~~  
2148 ~~decision is to maintain a consistent set of rules for mapping IPP to possibly many communication~~  
2149 ~~layers, even where URLs are not used as the addressing mechanism.~~  
2150 ~~2. Even though these two URLs might not be literally identical (one being relative and the other being~~  
2151 ~~absolute), they must both reference the same IPP object.~~  
2152 ~~3. The URL in the HTTP layer is either relative or absolute and is used by the HTTP server to route~~  
2153 ~~the HTTP request to the correct resource relative to that HTTP server. The HTTP server need~~  
2154 ~~not be aware of the URL within the operation request.~~  
2155 ~~4. Once the HTTP server resource begins to process the HTTP request, it might get the reference to~~  
2156 ~~the appropriate IPP Printer object from either the HTTP URL (using to the context of the HTTP~~  
2157 ~~server for relative URLs) or from the URL within the operation request; the choice is up to the~~  
2158 ~~implementation.~~  
2159 ~~5. HTTP URLs can be relative or absolute, but the target URL in the operation MUST be an absolute~~  
2160 ~~URL.~~

#### 2161 4.1.84.1.6 'uriScheme'

2162 The 'uriScheme' attribute syntax is a sequence of characters representing a URI scheme according to RFC  
2163 1738 [RFC1738]. Though RFC 1736 requires that the values be case-insensitive, IPP requires all lower  
2164 case to simplify comparing by IPP clients and Printer objects. Standard values for this syntax type are the  
2165 following keywords:

- 2166 'http': for HTTP schemed URIs (e.g., "http:...")  
2167 'https': for use with non-standard HTTPS schemed URIs (e.g., "https:...")  
2168 'ftp': for FTP schemed URIs (e.g., "ftp:...")  
2169 'mailto': for SMTP schemed URIs (e.g., "mailto:...")  
2170 'file': for file schemed URIs (e.g., "file:...")

2171  
2172 A Printer object MAY support any URI scheme that has been registered with IANA [IANA-MT]. The  
2173 maximum length of URI schemes used within IPP is 63 octets.

#### 2174 4.1.94.1.7 'charset'

2175 The 'charset' attribute syntax is a standard identifier for a charset. A charset is a coded character set and  
2176 encoding scheme. Charsets are used for labeling certain document contents and 'text' and 'name' attribute  
2177 values. The syntax and semantics of this attribute syntax are specified in RFC 2046 [RFC2046] and  
2178 contained in the IANA character-set Registry [IANA-CS] according to the IANA procedures [IANA-  
2179 CSa]. Though RFC 2046 requires that the values be case-insensitive US-ASCII, IPP requires all lower  
2180 case to simplify comparing by IPP clients and Printer objects. When a character-set in the IANA registry

2181 has more than one name (alias), the name labeled as "(preferred MIME name)", if present, SHALL be  
2182 used.

2183 The maximum length of charset values used within IPP is 63 octets.

2184 Some examples are:

2185 'utf-8': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as the UTF-8  
2186 [RFC2044279] transfer encoding scheme in which US-ASCII is a subset charset. ~~The 'utf-8'~~  
2187 ~~charset value supplied in the "attributes-charset" operation attribute (see Section 3.1.3), which is~~  
2188 ~~used to identify the charset of 'text' and 'name' attributes, SHALL be restricted to any characters~~  
2189 ~~defined by ISO 10646 [ISO10646-1].~~

2190 'us-ascii': 7-bit American Standard Code for Information Interchange (ASCII), ANSI X3.4-1986  
2191 [ASCII]. That standard defines US-ASCII, but RFC 2045 [46] eliminates most of the control  
2192 characters from conformant usage in MIME and IPP.

2193 'iso-8859-1': 8-bit One-Byte Coded Character Set, Latin Alphabet Nr 1 [ISO8859-1]. That standard  
2194 defines a coded character set that is used by Latin languages in the Western Hemisphere and  
2195 Western Europe. US-ASCII is a subset charset.

2196 'iso-10646-ucs-2': ISO 10646 Universal Multiple-Octet Coded Character Set (UCS) represented as  
2197 two octets (UCS-2), with the high order octet of each pair coming first (so-called Big Endian  
2198 integer).

2199  
2200 Some attribute descriptions MAY place additional requirements on charset values that may be used, such  
2201 as MANDATORY values that MUST be supported or additional restrictions, such as requiring that the  
2202 charset have US-ASCII as a subset charset.

#### 2203 ~~4.1.104.1.8~~ 'naturalLanguage'

2204 The 'naturalLanguage' attribute syntax is a standard identifier for a natural language and optionally a  
2205 country. The values for this syntax type are defined by RFC 1766 [RFC1766]. Though RFC 1766  
2206 requires that the values be case-insensitive US-ASCII, IPP requires all lower case to simplify comparing  
2207 by IPP clients and Printer objects. Examples include:

2208 'en': for English

2209 'en-us': for US English

2210 'fr': for French

2211 'de': for German

2212

2213 The maximum length of naturalLanguage values used within IPP is 63 octets.



2214 4.1.114.1.9 'mimeType'

2215 The 'mimeType' attribute syntax is the Internet Media Type (sometimes called MIME type) as  
2216 defined by RFC 2046 [RFC2046] and registered according to the procedures of RFC 2048 [RFC2048]  
2217 for identifying a document format. The value MAY include a charset parameter, depending on the  
2218 specification of the Media Type in the IANA Registry [IANA-MT]. Although most other IPP syntax  
2219 types allow for only lower-cased values, this syntax type allows for mixed-case values.

2220 Examples are:

2221 'text/html': An HTML document

2222 'text/plain': A plain text document in US-ASCII (RFC 2046 indicates that in the absence of the  
2223 charset parameter SHALL mean US-ASCII rather than simply unspecified) [RFC2046].

2224 'text/plain; charset=US-ASCII': A plain text document in US-ASCII [52, 56].

2225 'text/plain; charset=ISO-8859-1': A plain text document in ISO 8859-1 (Latin 1) [ISO8859-1].

2226 'text/plain; charset=utf-8': A plain text document in ISO 10646 represented as UTF-8 [RFC2044]

2227 'text/plain, charset=iso-10646-ucs-2': A plain text document in ISO 10646 represented in two octets  
2228 (UCS-2) [ISO10646-1]

2229 'application/postscript': A PostScript document [RFC2046]

2230 'application/vnd.hp-PCL': A PCL document [IANA-MT] (charset escape sequence embedded in the  
2231 document data)

2232 'application/octet-stream': Auto-sense - see below

2233

2234 One special type is 'application/octet-stream'. If the Printer object supports this value, the Printer object  
2235 SHALL be capable of auto-sensing the format of the document data. If the Printer object's default value  
2236 attribute "document-format-default" is set to 'application/octet-stream', the Printer object not only  
2237 supports auto-sensing of the document format, but will depend on the result of applying its auto-sensing  
2238 when the client does not supply the "document-format" attribute. If the client supplies a document  
2239 format value, the Printer SHALL rely on the supplied attribute, rather than trust its auto-sensing  
2240 algorithm. To summarize:

- 2241 1. If the client does not supply a document format value, the Printer MUST rely on its default value  
2242 setting (which may be 'application/octet-stream' indicating an auto-sensing mechanism).
- 2243 2. If the client supplies a value other than 'application/octet-stream', the client is supplying valid  
2244 information about the format of the document data and the Printer object SHALL trust the client  
2245 supplied value more than the outcome of applying an automatic format detection mechanism. For  
2246 example, the client may be requesting the printing of a PostScript file as a 'text/plain' document.  
2247 The Printer object SHALL print a text representation of the PostScript commands rather than  
2248 interpret the stream of PostScript commands and print the result.



2249 3. If the client supplies a value of 'application/octet-stream', the client is indicating that the Printer  
2250 object SHALL use its auto-sensing mechanism on the client supplied document data whether  
2251 auto-sensing is the Printer object's default or not.  
2252

2253 Note: Since the auto-sensing algorithm is probabilistic, if the client requests both auto-sensing  
2254 ("document-format" set to 'application/octet-stream') and true fidelity ("ipp-attribute-fidelity" set to  
2255 'true'), the Printer object might not be able to guarantee exactly what the end user intended (the auto-  
2256 sensing algorithm might mistake one document format for another ), but it is able to guarantee that its  
2257 auto-sensing mechanism be used.

2258 The maximum length of a 'mimeType' value in IPP is ~~63~~255 octets.

2259 ~~4.1.124~~4.1.10 'octetString'

2260 The 'octetString' attribute syntax is a sequence of octets encoded in a maximum of 1023 octets which is  
2261 indicated in sub-section headers using the notation: octetString(MAX). This syntax type is used for  
2262 opaque data.

2263 ~~4.1.134~~4.1.11 'boolean'

2264 The 'boolean' attribute syntax is similar to an enum with only two values: 'true' and 'false'.

2265 ~~4.1.144~~4.1.12 'integer'

2266 The 'integer' attribute syntax is an integer value that is in the range from  $-2^{31}$  (MIN) to  $2^{31} - 1$   
2267 (MAX). Each individual attribute may specify the range constraint explicitly in sub-section headers if the  
2268 range is different from the full range of possible integer values. For example: job-priority  
2269 (integer(1:100)) for the "job-priority" attribute. However, the enforcement of that additional constraint is  
2270 up to the IPP objects, not the protocol.

2271 ~~4.1.154~~4.1.13 'rangeOfInteger'

2272 The 'rangeOfInteger' attribute syntax is an ordered pair of integers that defines an inclusive range of  
2273 integer values. The first integer specifies the lower bound and the second specifies the upper bound. If a  
2274 range constraint is specified in the header description for an attribute in this document whose attribute  
2275 syntax is 'rangeOfInteger' (i.e., 'X:Y' indicating X as a minimum value and Y as a maximum value), then  
2276 the constraint applies to both integers.

2277 ~~4.1.164.1.14~~ 'dateTime'

2278 The 'dateTime' attribute syntax is a standard, fixed length, 11 octet representation of the "DateAndTime"  
2279 syntax as defined in RFC 1903 [RFC1903]. RFC 1903 also identifies an 8 octet representation of a  
2280 "DateAndTime" value, but IPP objects MUST use the 11 octet representation. A user interface will  
2281 provide a mapping between protocol dateTime values and displayable user-friendly words or presentation  
2282 values and phrases which are localized to the natural language and date format of the user.

2283 ~~4.1.174.1.15~~ 'resolution'

2284 The 'resolution' attribute syntax specifies a two-dimensional resolution in the indicated units. It consists  
2285 of 3 integers: a cross feed direction resolution (positive integer value), a feed direction resolution  
2286 (positive integer value), and a units value. The semantics of these three components are taken from the  
2287 Printer MIB [RFC1759] suggested values. That is, the cross feed direction component resolution  
2288 component is the same as the prtMarkerAddressabilityXFeedDir object in the Printer MIB, the feed  
2289 direction component resolution component is the same as the prtMarkerAddressabilityFeedDir in the  
2290 Printer MIB, and the units component is the same as the prtMarkerAddressabilityUnit object in the  
2291 Printer MIB (namely, '3' indicates dots per inch and '4' indicates dots per centimeter). All three values  
2292 MUST be present even if the first two values are the same. Example: '300', '600', '3' indicates a 300 dpi  
2293 cross-feed direction resolution, a 600 dpi feed direction resolution, since a '3' indicates dots per inch  
2294 (dpi).

2295 ~~4.1.184.1.16~~ '1setOf X'

2296 The '1setOf X' attribute syntax is 1 or more values of attribute syntax type X. This syntax type is used  
2297 for multi-valued attributes. The syntax type is called '1setOf' rather than just 'setOf' as a reminder that  
2298 the set of values SHALL NOT be empty (i.e., a set of size 0). Sets are normally unordered. However  
2299 each attribute description of this type may specify that the values MUST be in a certain order for that  
2300 attribute.

## 2301 4.2 Job Template Attributes

2302 Job Template attributes describe job processing behavior. Support for Job Template attributes by a  
2303 Printer object is OPTIONAL (see section 13.2.3 for a description of support for OPTIONAL attributes).  
2304 Also, clients OPTIONALLY supply Job Template attributes in create requests.

2305 Job Template attributes conform to the following rules. For each Job Template attribute called "xxx":

- 2306 1. If the Printer object supports "xxx" then it SHALL support both a "xxx-default" attribute (unless  
2307 there is a "No" in the table below) and a "xxx-supported" attribute. If the Printer object doesn't

2308 support "xxx", then it SHALL support neither an "xxx-default" attribute nor an "xxx-supported"  
2309 attribute, and it SHALL treat an attribute "xxx" supplied by a client as unsupported. An attribute  
2310 "xxx" may be supported for some document formats and not supported for other document  
2311 formats. For example, it is expected that a Printer object would only support "orientation-  
2312 requested" for some document formats (such as 'text/plain' or 'text/html') but not others (such as  
2313 'application/postscript').

2314

2315 2. "xxx" is OPTIONALLY supplied by the client in a create request. If "xxx" is supplied, the client is  
2316 indicating a desired job processing behavior for this Job. When "xxx" is not supplied, the client is  
2317 indicating that the Printer object apply its default job processing behavior at job processing time if  
2318 the document content does not contain an embedded instruction indicating an xxx-related  
2319 behavior.

2320

2321 Note: Since an administrator MAY change the default value attribute after a Job object has been  
2322 submitted but before it has been processed, the default value used by the Printer object at job  
2323 processing time may be different that the default value in effect at job submission time.

2324

2325 3. The "xxx-supported" attribute is a Printer object attribute that describes which job processing  
2326 behaviors are supported by that Printer object. A client can query the Printer object to find out  
2327 what xxx-related behaviors are supported by inspecting the returned values of the "xxx-  
2328 supported" attribute.

2329

2330 Note: The "xxx" in each "xxx-supported" attribute name is singular, even though an "xxx-  
2331 supported" attribute usually has more than one value, such as "job-sheet-supported", unless the  
2332 "xxx" Job Template attribute is plural, such as "finishings" or "sides". In such cases the "xxx-  
2333 supported" attribute names are: "finishings-supported" and "sides-supported".

2334

2335 4. The "xxx-default" default value attribute describes what will be done at job processing time when  
2336 no other job processing information is supplied by the client (either explicitly as an IPP attribute in  
2337 the create request or implicitly as an embedded instruction within the document data).

2338

2339 If an application wishes to present an end user with a list of supported values from which to choose, the  
2340 application SHOULD query the Printer object for its supported value attributes. The application  
2341 SHOULD also query the default value attributes. If the application then limits selectable values to only  
2342 those value that are supported, the application can guarantee that the values supplied by the client in the  
2343 create request all fall within the set of supported values at the Printer. When querying the Printer, the  
2344 client MAY enumerate each attribute by name in the Get-Printer-Attributes Request, or the client MAY  
2345 just name the "job-template" group in order to get the complete set of supported attributes (both  
2346 supported and default attributes).

2347 The "finishings" attribute is an example of a Job Template attribute. It can take on a set of values such as  
2348 'staple', 'punch', and/or 'cover'. A client can query the Printer object for the "finishings-supported"  
2349 attribute and the "finishings-default" attribute. The supported attribute contains a set of supported  
2350 values. The default value attribute contains the finishing value(s) that will be used for a new Job if the  
2351 client does not supply a "finishings" attribute in the create request and the document data does not  
2352 contain any corresponding finishing instructions. If the client does supply the "finishings" attribute in the  
2353 create request, the IPP object validates the value or values to make sure that they are a subset of the  
2354 supported values identified in the Printer object's "finishings-supported" attribute. See section 3.2.1.2.

2355 The table below summarizes the names and relationships for all Job Template attributes. The first column  
2356 of the table (labeled "Job Attribute") shows the name and syntax for each Job Template attribute in the  
2357 Job object. These are the attributes that can optionally be supplied by the client in a create request. The  
2358 last two columns (labeled "Printer: Default Value Attribute" and "Printer: Supported Values Attribute")  
2359 shows the name and syntax for each Job Template attribute in the Printer object (the default value  
2360 attribute and the supported values attribute). A "No" in the table means the Printer SHALL NOT  
2361 support the attribute (that is, the attribute is simply not applicable). For brevity in the table, the 'text' and  
2362 'name' entries do not show the maximum length, as in "(127)". (MAX).

2363	+	=====+	=====+	=====+
2364		Job Attribute	Printer: Default Value	Printer: Supported
2365			Attribute	Values Attribute
2366		=====+	=====+	=====+
2367		job-priority	job-priority-default	job-priority-supported
2368		(integer 1:100)	(integer 1:100)	(integer 1:100)
2369		-----+	-----+	-----+
2370		job-hold-until	job-hold-until-	job-hold-until-
2371		(type3 keyword	default	supported
2372		name)	(type3 keyword	(1setOf
2373			name)	type3 keyword   name)
2374		-----+	-----+	-----+
2375		job-sheets	job-sheets-default	job-sheets-supported
2376		(type3 keyword	(type3 keyword	(1setOf
2377		name)	name)	type3 keyword   name)
2378		-----+	-----+	-----+
2379		multiple-document-	multiple-document-	multiple-document-
2380		handling	handling-default	handling-supported
2381		(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2382		-----+	-----+	-----+
2383		copies	copies-default	copies-supported
2384		(integer (1:MAX))	(integer (1:MAX))	(rangeOfInteger
2385				(1:MAX))
2386		-----+	-----+	-----+
2387		finishings	finishings-default	finishings-supported
2388		(1setOf type2 enum)	(1setOf type2 enum)	(1setOf type2 enum)
2389		-----+	-----+	-----+
2390		page-ranges	No	page-ranges-
2391		(1setOf		supported (boolean)
2392		rangeOfInteger		
2393		(1:MAX))		
2394		-----+	-----+	-----+
2395		sides	sides-default	sides-supported
2396		(type2 keyword)	(type2 keyword)	(1setOf type2 keyword)
2397		-----+	-----+	-----+
2398		number-up	number-up-default	number-up-supported
2399		(integer (1:MAX))	(integer (1:MAX))	(1setOf integer
2400				(1:MAX)
2401				rangeOfInteger
2402				(1:MAX))
2403		-----+	-----+	-----+
2404		orientation-	orientation-requested-	orientation-requested-
2405		requested	default	supported
2406		(type2 enum)	(type2 enum)	(1setOf type2 enum)
2407		-----+	-----+	-----+

2408	media	media-default	media-supported
2409	(type3 keyword	(type3 keyword	(1setOf
2410	name)	name)	type3 keyword   name)
2411			
2412			media-ready
2413			(1setOf
2414			type3 keyword   name)
2415	+-----+	+-----+	+-----+
2416	printer-resolution	printer-resolution-	printer-resolution-
2417	(resolution)	default	supported
2418		(resolution)	(1setOf resolution)
2419	+-----+	+-----+	+-----+
2420	print-quality	print-quality-default	print-quality-
2421	(type2 enum)	(type2 enum)	supported
2422			(1setOf type2 enum)
2423	+-----+	+-----+	+-----+
2424			
2425			

#### 2426 4.2.1 job-priority (integer(1:100))

2427 This attribute specifies a priority for scheduling the Job. A higher value specifies a higher priority. The  
 2428 value 1 indicates the lowest possible priority. The value 100 indicates the highest possible priority.  
 2429 Among those jobs that are ready to print, a Printer SHALL print all jobs with a priority value of n before  
 2430 printing those with a priority value of n-1 for all n.

2431 If the Printer object supports this attribute, it SHALL always support the full range from 1 to 100. No  
 2432 administrative restrictions are permitted. This way an end-user can always make full use of the entire  
 2433 range with any Printer object. If privileged jobs are implemented outside IPP/1.0, they SHALL have  
 2434 priorities higher than 100, rather than restricting the range available to end-users.

2435 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
 2436 object SHALL use the value of the Printer object's "job-priority-default" at job submission time (unlike  
 2437 most Job Template attributes that are used if necessary at job processing time).

2438 The syntax for the "job-priority-supported" is also integer(1:100). This single integer value indicates the  
 2439 number of priority levels supported. The Printer object SHALL take the value supplied by the client and  
 2440 map it to the closest integer in a sequence of n integers values that are evenly distributed over the range  
 2441 from 1 to 100 using the formula:

$$2442 \quad \text{roundToNearestInt}((100x+50)/n)$$

2443 where n is the value of "job-priority-supported" and x ranges from 0 through n-1.

2444 For example, if n=1 the sequence of values is 50; if n=2, the sequence of values is: 25 and 75; if n = 3,  
2445 the sequence of values is: 17, 50 and 83; if n = 10, the sequence of values is: 5, 15, 25, 35, 45, 55, 65,  
2446 75, 85, and 95; if n = 100, the sequence of values is: 1, 2, 3, ... 100.

2447 If the value of the Printer object's "job-priority-supported" is 10 and the client supplies values in the range  
2448 1 to 10, the Printer object maps them to 5, in the range 11 to 20, the Printer object maps them to 15, etc.

#### 2449 4.2.2 job-hold-until (type3 keyword | name (MAX))

2450 This attribute specifies the named time period during which the Job SHALL become a candidate for  
2451 printing.

2452 Standard values for named time periods are:

2453 'no-hold': immediately, if there are not other reasons to hold the job

2454 'day-time': during the day

2455 'evening': evening

2456 'night': night

2457 'weekend': weekend

2458 'second-shift': second-shift (after close of business)

2459 'third-shift': third-shift (after midnight)

2460

2461 An administrator SHALL associate allowable print times with a named time period (by means outside  
2462 IPP/1.0). An administrator is encouraged to pick names that suggest the type of time period. An  
2463 administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending on  
2464 implementation.

2465 If the value of this attribute specifies a time period that is in the future, the Printer SHALL add the 'job-  
2466 hold-until-specified' value to the job's "job-state-reasons" attribute, move the job to the 'pending-held'  
2467 state, and SHALL NOT schedule the job for printing until the specified time-period arrives. When the  
2468 specified time period arrives, the Printer SHALL remove the 'job-hold-until-specified' value from the job's  
2469 "job-state-reason" attribute and, if there are no other job state reasons that keep the job in the 'pending-  
2470 held' state, the Printer SHALL consider the job as a candidate for processing by moving the job to the  
2471 'pending' state.

2472 If this job attribute value is the named value 'no-hold', or the specified time period has already started, the  
2473 job SHALL be a candidate for processing immediately.

2474 If the client does not supply this attribute and this attribute is supported by the Printer object, the Printer  
2475 object SHALL use the value of the Printer object's "job-hold-until-default" at job submission time (unlike  
2476 most Job Template attributes that are used if necessary at job processing time).

## 2477 4.2.3 job-sheets (type3 keyword | name(MAX))

2478 This attribute determines which job start/end sheet(s), if any, SHALL be printed with a job.

2479 Standard values are:

2480 'none': no job sheet is printed

2481 'standard': one or more site specific standard job sheets are printed, e.g. a single start sheet or both  
2482 start and end sheet is printed

2483

2484 An administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending  
2485 on implementation.

2486 Note: The effect of this attribute on jobs with multiple documents MAY be affected by the "multiple-  
2487 document-handling" job attribute (section 4.2.4), depending on the job sheet semantics.

## 2488 4.2.4 multiple-document-handling (type2 keyword)

2489 This attribute is relevant only if a job consists of two or more documents. The attribute controls finishing  
2490 operations and the placement of one or more print-stream pages into impressions and onto media sheets.  
2491 When the value of the "copies" attribute exceeds 1, it also controls the order in which the copies that  
2492 result from processing the documents are produced. For the purposes of this explanations, if "a"  
2493 represents an instance of document data, then the result of processing the data in document "a" is a  
2494 sequence of media sheets represented by "a(\*)".

2495 Standard values are:

2496 'single-document': If a Job object has multiple documents, say, the document data is called a and b,  
2497 then the result of processing all the document data (a and then b) SHALL be treated as a single  
2498 sequence of media sheets for finishing operations; that is, finishing would be performed on the  
2499 concatenation of the sequences a(\*),b(\*). The Printer object SHALL NOT force the data in each  
2500 document instance to be formatted onto a new print-stream page, nor to start a new impression  
2501 on a new media sheet. If more than one copy is made, the ordering of the sets of media sheets  
2502 resulting from processing the document data SHALL be a(\*), b(\*), a(\*), b(\*), ..., and the Printer  
2503 object SHALL force each copy (a(\*),b(\*)) to start on a new media sheet.

2504 'separate-documents-uncollated-copies': If a Job object has multiple documents, say, the document  
2505 data is called a and b, then the result of processing the data in each document instance SHALL be  
2506 treated as a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*)  
2507 would each be finished separately. The Printer object SHALL force each copy of the result of  
2508 processing the data in a single document to start on a new media sheet. If more than one copy is



2509           made, the ordering of the sets of media sheets resulting from processing the document data  
2510           SHALL be a(\*), a(\*), ..., b(\*), b(\*) ... .  
2511       'separate-documents-collated-copies': If a Job object has multiple documents, say, the document data  
2512           is called a and b, then the result of processing the data in each document instance SHALL be  
2513           treated as a single sequence of media sheets for finishing operations; that is, the sets a(\*) and b(\*)  
2514           would each be finished separately. The Printer object SHALL force each copy of the result of  
2515           processing the data in a single document to start on a new media sheet. If more than one copy is  
2516           made, the ordering of the sets of media sheets resulting from processing the document data  
2517           SHALL be a(\*), b(\*), a(\*), b(\*), ... .

2518  
2519       The 'single-document' value is the same as 'separate-documents-collated-copies' with respect to ordering  
2520       of print-stream pages, but not media sheet generation, since 'single-document' will put the first page of  
2521       the next document on the back side of a sheet if an odd number of pages have been produced so far for  
2522       the job, while 'separate-documents-collated-copies' always forces the next document or document copy  
2523       on to a new sheet. In addition, if the "finishings" attribute specifies 'staple', then with 'single-document',  
2524       documents a and b are stapled together as a single document, but with 'separate-documents-uncollated-  
2525       copies' and 'separate-documents-collated-copies', documents a and b are stapled separately.

2526       Note: None of these values provide means to produce uncollated sheets within a document, i.e., where  
2527       multiple copies of sheet n are produced before sheet n+1 of the same document.

2528       The relationship of this attribute and the other attributes that control document processing is described in  
2529       section 16.5.

#### 2530       4.2.5 copies (integer(1:MAX))

2531       This attribute specifies the number of copies to be printed.

2532       On many devices the supported number of collated copies will be limited by the number of physical  
2533       output bins on the device, and may be different from the number of uncollated copies which can be  
2534       supported.

2535       Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2536       document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2537       attributes that control document processing is described in section 16.5.

#### 2538       4.2.6 finishings (1setOf type2 enum)

2539       This attribute identifies the finishing operations that the Printer uses for each copy of each printed  
2540       document in the Job. For Jobs with multiple documents, the "multiple-document-handling" attribute  
2541       determines what constitutes a "copy" for purposes of finishing.

2542 Standard values are:

2543	Value	Symbolic Name and Description
2544		
2545	'3'	'none': Perform no finishing
2546	'4'	'staple': Bind the document(s) with one or more staples. The exact number and placement
2547		of the staples is site-defined.
2548	'5'	'punch': This value indicates that holes are required in the finished document. The exact
2549		number and placement of the holes is site-defined. The punch specification MAY
2550		be satisfied (in a site- and implementation-specific manner) either by
2551		drilling/punching, or by substituting pre-drilled media.
2552	'6'	'cover': This value is specified when it is desired to select a non-printed (or pre-printed)
2553		cover for the document. This does not supplant the specification of a printed cover
2554		(on cover stock medium) by the document itself.
2555	'7'	'bind': This value indicates that a binding is to be applied to the document; the type and
2556		placement of the binding is site-defined."
2557		

2558 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2559 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2560 attributes that control document processing is described in section 16.5.

2561 If the client supplies a value of 'none' along with any other combination of values, it is the same as if only  
2562 that other combination of values had been supplied (that is the 'none' value has no effect).

#### 2563 4.2.7 page-ranges (1setOf rangeOfInteger (1:MAX))

2564 This attribute identifies the range(s) of print-stream pages that the Printer object uses for each copy of  
2565 each document which are to be printed. Nothing is printed for any pages identified that do not exist in  
2566 the document(s). Ranges SHALL be in ascending order, for example: 1-3, 5-7, 15-19 and SHALL NOT  
2567 overlap, so that a non-spooling Printer object can process the job in a single pass. If the ranges are not  
2568 ascending or are overlapping, the IPP object SHALL reject the request and return the 'client-error-bad-  
2569 request' status code. The attribute is associated with print-stream pages not application-numbered pages  
2570 (for example, the page numbers found in the headers and or footers for certain word processing  
2571 applications).

2572 For Jobs with multiple documents, the "multiple-document-handling" attribute determines what  
2573 constitutes a "copy" for purposes of the specified page range(s). When "multiple-document-handling" is  
2574 'single-document', the Printer object SHALL apply each supplied page range once to the concatenation of  
2575 the print-stream pages. For example, if there are 8 documents of 10 pages each, the page-range '41:60'  
2576 prints the pages in the 5th and 6th documents as a single document and none of the pages of the other  
2577 documents are printed. When "multiple-document-handling" is 'separate-document-uncollated-copies' or

2578 'separate-document-collated-copies', the Printer object SHALL apply each supplied page range repeatedly  
2579 to each document copy. For the same job, the page-range '1:3, 10:10' would print the first 3 pages and  
2580 the 10th page of each of the 8 documents in the Job, as 8 separate documents.

2581 In most cases, the exact pages to be printed will be generated by a device driver and this attribute would  
2582 not be required. However, when printing an archived document which has already been formatted, the  
2583 end user may elect to print just a subset of the pages contained in the document. In this case, if page-  
2584 range = n.m is specified, the first page to be printed will be page n. All subsequent pages of the document  
2585 will be printed through and including page m.

2586 "page-ranges-supported" is a boolean value indicating whether or not the printer is capable of supporting  
2587 the printing of page ranges. This capability may differ from one PDL to another. There is no "page-  
2588 ranges-default" attribute. If the "page-ranges" attribute is not supplied by the client, all pages of the  
2589 document will be printed.

2590 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2591 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2592 attributes that control document processing is described in section 16.5.

#### 2593 4.2.8 sides (type2 keyword)

2594 This attribute specifies how print-stream pages are to be imposed upon the sides of an instance of a  
2595 selected medium, i.e., an impression.

2596 The standard values are:

2597 'one-sided': imposes each consecutive print-stream page upon the same side of consecutive media  
2598 sheets.

2599 'two-sided-long-edge': imposes each consecutive pair of print-stream pages upon front and back sides  
2600 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the  
2601 medium would be correct for the reader as if for binding on the long edge. This imposition is  
2602 sometimes called 'duplex' or 'head-to-head'.

2603 'two-sided-short-edge': imposes each consecutive pair of print-stream pages upon front and back sides  
2604 of consecutive media sheets, such that the orientation of each pair of print-stream pages on the  
2605 medium would be correct for the reader as if for binding on the short edge. This imposition is  
2606 sometimes called 'tumble' or 'head-to-toe'.

2607  
2608 'two-sided-long-edge', 'two-sided-short-edge', 'tumble', and 'duplex' all work the same for portrait or  
2609 landscape. However 'head-to-toe' is 'tumble' in portrait but 'duplex' in landscape. 'head-to-head' also  
2610 switches between 'duplex' and 'tumble' when using portrait and landscape modes.

2611 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
 2612 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
 2613 attributes that control document processing is described in section 16.5.

#### 2614 4.2.9 number-up (integer(1:MAX))

2615 This attribute specifies the number of print-stream pages to impose upon a single side of an instance of a  
 2616 selected medium. For example, if the value is

2617	Value	Description
2618		
2619	'1'	The Printer SHALL place one print-stream page on a single side of an instance of the
2620		selected medium (MAY add some sort of translation, scaling, or rotation).
2621	'2'	The Printer SHALL place two print-stream pages on a single side of an instance of the
2622		selected medium (MAY add some sort of translation, scaling, or rotation).
2623	'4'	The Printer SHALL place four print-stream pages on a single side of an instance of the
2624		selected medium (MAY add some sort of translation, scaling, or rotation).

2625  
 2626 This attribute primarily controls the translation, scaling and rotation of print-stream pages.

2627 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
 2628 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
 2629 attributes that control document processing is described in section 16.5.

#### 2630 4.2.10 orientation-requested (type2 enum)

2631 This attribute indicates the desired orientation for printed print-stream pages; it does not describe the  
 2632 orientation of the client-supplied print-stream pages.

2633 For some document formats (such as 'application/postscript'), the desired orientation of the print-stream  
 2634 pages is specified within the document data. This information is generated by a device driver prior to the  
 2635 submission of the print job. Other document formats (such as 'text/plain') do not include the notion of  
 2636 desired orientation within the document data. In the latter case it is possible for the Printer object to bind  
 2637 the desired orientation to the document data after it has been submitted. It is expected that a Printer  
 2638 object would only support "orientations-requested" for some document formats (e.g., 'text/plain' or  
 2639 'text/html') but not others (e.g., 'application/postscript'). This is no different than any other Job Template  
 2640 attribute since section 4.2, item 1, points out that a Printer object may support or not support any Job  
 2641 Template attribute based on the document format supplied by the client. However, a special mention is  
 2642 made here since it is very likely that a Printer object will support "orientation-requested" for only a subset  
 2643 of the supported document formats.

2644 Standard values are:

2645	Value	Symbolic Name and Description
2646		
2647	'3'	'portrait': The content will be imaged across the short edge of the medium.
2648	'4'	'landscape': The content will be imaged across the long edge of the medium. Landscape is
2649		defined to be a rotation of the print-stream page to be imaged by +90 degrees with
2650		respect to the medium (i.e. anti-clockwise) from the portrait orientation. Note:
2651		The +90 direction was chosen because simple finishing on the long edge is the
2652		same edge whether portrait or landscape
2653	'5'	'reverse-landscape': The content will be imaged across the long edge of the medium.
2654		Reverse-landscape is defined to be a rotation of the print-stream page to be imaged
2655		by -90 degrees with respect to the medium (i.e. clockwise) from the portrait
2656		orientation. Note: The 'reverse-landscape' value was added because some
2657		applications rotate landscape -90 degrees from portrait, rather than +90 degrees.
2658	'6'	'reverse-portrait': The content will be imaged across the shshort edge of the medium.
2659		Reverse-portrait is defined to be a rotation of the print-stream page to be imaged
2660		by 180 degrees with respect to the medium from the portrait orientation. Note:
2661		The 'reverse-portrait' value was added for use with the "finishings" attribute in
2662		cases where the opposite edge is desired for finishing a portrait document on
2663		simple finishing devices that have only one finishing position. Thus a 'text/plain'
2664		portrait document can be stapled "on the right" by a simple finishing device as is
2665		common use with some middle eastern languages such as Hebrew.
2666		

2667 Note: The effect of this attribute on jobs with multiple documents is controlled by the "multiple-  
2668 document-handling" job attribute (section 4.2.4) and the relationship of this attribute and the other  
2669 attributes that control document processing is described in section 16.5.

#### 2670 4.2.11 media (type3 keyword | name(MAX))

2671 This attribute identifies the medium that the Printer uses for all impressions of the Job.

2672 The values for "media" include medium-names, medium-sizes, input-trays and electronic forms so that  
2673 one attribute specifies the media. If a Printer object supports a medium name as a value of this attribute,  
2674 such a medium name implicitly selects an input-tray that contains the specified medium. If a Printer  
2675 object supports a medium size as a value of this attribute, such a medium size implicitly selects a medium  
2676 name that in turn implicitly selects an input-tray that contains the medium with the specified size. If a  
2677 Printer object supports an input-tray as the value of this attribute, such an input-tray implicitly selects the  
2678 medium that is in that input-tray at the time the job prints. This case includes manual-feed input-trays. If  
2679 a Printer object supports an electronic form as the value of this attribute, such an electronic form

2680 implicitly selects a medium-name that in turn implicitly selects an input-tray that contains the medium  
 2681 specified by the electronic form. The electronic form also implicitly selects an image that the Printer  
 2682 SHALL merge with the document data as its prints each page.

2683 Standard values are (taken from ISO DPA and the Printer MIB) and are listed in section 15. An  
 2684 administrator MAY define additional values using the 'name' or 'keyword' attribute syntax, depending on  
 2685 implementation.

2686 There is also an additional Printer attribute named "media-ready" which differs from "media-supported" in  
 2687 that legal values only include the subset of "media-supported" values that are physically loaded and ready  
 2688 for printing with no operator intervention required. If an IPP object supports "media-supported", it  
 2689 NEED NOT support "media-ready".

2690 The relationship of this attribute and the other attributes that control document processing is described in  
 2691 section 16.5.

#### 2692 4.2.12 printer-resolution (resolution)

2693 This attribute identifies the resolution that Printer uses for the Job.

#### 2694 4.2.13 print-quality (type2 enum)

2695 This attribute specifies the print quality that the Printer uses for the Job.

2696 The standard values are:

2697	Value	Symbolic Name and Description
2698		
2699	'3'	'draft': lowest quality available on the printer
2700	'4'	'normal': normal or intermediate quality on the printer
2701	'5'	'high': highest quality available on the printer
2702		

### 2703 4.3 Job Description Attributes

2704 The attributes in this section form the attribute group called "job-description". The following table  
 2705 summarizes these attributes. The third column indicates whether the attribute is a MANDATORY  
 2706 attribute that MUST be supported by Printer objects. If it is not indicated as MANDATORY, then it is  
 2707 OPTIONAL. The maximum size in octets for 'text' and 'name' attributes is indicated in parentheses.

	Attribute	Syntax	MANDATORY?
2708			
2709			
2710			
2711	job-uri	uri	MANDATORY
2712			
2713	job-id	integer(1:MAX)	MANDATORY
2714			
2715	job-printer-uri	uri	MANDATORY
2716			
2717	job-more-info	uri	
2718			
2719	job-name	name (MAX)	MANDATORY
2720			
2721	job-originating-user-name	name (MAX)	MANDATORY
2722			
2723	job-state	type1 enum	MANDATORY
2724			
2725	job-state-reasons	1setOf type2 keyword	
2726			
2727	job-state-message	text (MAX)	
2728			
2729	number-of-documents	integer (0:MAX)	
2730			
2731	output-device-assigned	name (127)	
2732			
2733	time-at-creation	integer (0:MAX)	
2734			
2735	time-at-processing	integer (0:MAX)	
2736			
2737	time-at-completed	integer (0:MAX)	
2738			
2739	number-of-intervening-jobs	integer (0:MAX)	
2740			
2741	job-message-from-operator	text (127)	
2742			
2743	job-k-octets	integer (0:MAX)	
2744			
2745	job-impressions	integer (0:MAX)	
2746			
2747	job-media-sheets	integer (0:MAX)	
2748			
2749	job-k-octets-processed	integer (0:MAX)	
2750			
2751	job-impressions-completed	integer (0:MAX)	
2752			

2753	job-media-sheets-completed	integer (0:MAX)	
2754	+-----+-----+-----+		
2755	attributes-charset	charset	MANDATORY
2756	+-----+-----+-----+		
2757	attributes-natural-language	naturalLanguage	MANDATORY
2758	+-----+-----+-----+		
2759			
2760			

#### 2761 4.3.1 job-uri (uri)

2762 This MANDATORY attribute contains the URI for the job. The Printer object, on receipt of a new job,  
 2763 generates a URI which identifies the new Job. The Printer object returns the value of the "job-uri"  
 2764 attribute as part of the response to a create request. The precise format of a Job URI is implementation  
 2765 dependent. If the Printer object supports more than one URI and there is some relationship between the  
 2766 newly formed Job URI and the Printer object's URI, the Printer object uses the Printer URI supplied by  
 2767 the client in the create request. ~~however the URI MUST reference the access method that supports the~~  
 2768 ~~correct security characteristics.~~ For example, That is if the create request comes in over a secure  
 2769 channel, the new Job URI MUST use the same secure channel. This can guaranteed because the Printer  
 2770 object is responsible for generating ~~this the Job URI and the Printer object MUST is be~~ aware of its  
 2771 security configuration and policy as well as the Printer URI used in the create request.

2772 For a description of this attribute and its relationship to "job-id" and "job-printer-uri" attribute, see the  
 2773 discussion in section 2.4 on "Object Identity".

#### 2774 4.3.2 job-id (integer(1:MAX))

2775 This MANDATORY attribute contains the ID of the job. The Printer, on receipt of a new job, generates  
 2776 an ID which identifies the new Job on that Printer. The Printer returns the value of the "job-id" attribute  
 2777 as part of the response to a create request. The 0 value is not ~~used~~ included to allow for compatibility  
 2778 with SNMP index values which also cannot be 0.

2779 For a description of this attribute and its relationship to "job-uri" and "job-printer-uri" attribute, see the  
 2780 discussion in section 2.4 on "Object Identity".

#### 2781 4.3.3 job-printer-uri (uri)

2782 This MANDATORY attribute identifies the Printer object that created this Job object. When a Printer  
 2783 object creates a Job object, it populates this attribute with the Printer object URI that was used in the  
 2784 create request. This attribute permits a client to identify the Printer object that created this Job object  
 2785 when only the Job object's URI is available to the client. The client queries the creating Printer object to  
 2786 determine which languages, charsets, operations, are supported for this Job.



2787 For a description of this attribute and its relationship to "job-uri" and "job-id" attribute, see the discussion  
2788 in section 2.4 on "Object Identity".

#### 2789 4.3.4 job-more-info (uri)

2790 Similar to "printer-more-info", this attribute contains the URI referencing some resource with more  
2791 information about this Job object, perhaps an HTML page containing information about the Job.

#### 2792 4.3.5 job-name (name(MAX))

2793 This MANDATORY attribute is the name of the job. It is a name that is more user friendly than the "job-  
2794 uri" attribute value. It does not need to be unique between Jobs. The Job's "job-name" attribute is set to  
2795 the value supplied by the client in the "job-name" operation attribute in the create request (see Section  
2796 3.2.1.1). If, however, the "job-name" operation attribute is not supplied by the client in the create  
2797 request, the Printer object, on creation of the Job, SHALL generate a name. The printer SHOULD  
2798 generate the value of the Job's "job-name" attribute from the first of the following sources that produces a  
2799 value: 1) the "document-name" operation attribute of the first (or only) document, 2) the "document-  
2800 URI" attribute of the first (or only) document, or 3) any other piece of Job specific and/or Document  
2801 Content information.

#### 2802 4.3.6 job-originating-user-name (name(MAX))

2803 This MANDATORY attribute contains the name of the end user that submitted the print job. The Printer  
2804 object sets this attribute to the most authenticated printable name that it can obtain from the  
2805 authentication service over which the IPP operation was received. Only if such is not available, does the  
2806 Printer object use the value supplied by the client in the "requesting-user-name" operation attribute of the  
2807 create operation (see Section 8).

2808 Note: The Printer object needs to keep an internal originating user id of some form, typically as a  
2809 credential of a principal, with the Job object. Since such an internal attribute is implementation-  
2810 dependent and not of interest to clients, it is not specified as a Job Description attribute. This originating  
2811 user id is used for authorization checks (if any) on all subsequent operation.

#### 2812 4.3.7 job-state (type1 enum)

2813 This MANDATORY attribute identifies the current state of the job. Even though the IPP protocol  
2814 defines eight values for job states, implementations only need to support those states which are  
2815 appropriate for the particular implementation. In other words, a Printer supports only those job states  
2816 implemented by the output device and available to the Printer object implementation.

2817 Standard values are:

2818 Values Symbolic Name and Description

2819  
2820 '3' 'pending': The job is a candidate to start processing, but is not yet processing.

2821  
2822 '4' 'pending-held': The job is not a candidate for processing for any number of reasons but  
2823 will return to the 'pending' state as soon as the reasons are no longer present. The  
2824 job's "job-state-reason" attribute SHALL indicate why the job is no longer a  
2825 candidate for processing.

2826  
2827 '5' 'processing': One or more of:

- 2828
- 2829 1. the job is using, or is attempting to use, one or more purely software processes
  - 2830 that are analyzing, creating, or interpreting a PDL, etc.,
  - 2831 2. the job is using, or is attempting to use, one or more hardware devices that are
  - 2832 interpreting a PDL, making marks on a medium, and/or performing finishing, such
  - 2833 as stapling, etc.,
  - 2834 3. the Printer object has made the job ready for printing, but the output device is
  - 2835 not yet printing it, either because the job hasn't reached the output device or
  - 2836 because the job is queued in the output device or some other spooler, awaiting the
  - 2837 output device to print it.

2838

2839 When the job is in the 'processing' state, the entire job state includes the detailed  
2840 status represented in the printer's "printer-state", "printer-state-reasons", and  
2841 "printer-state-message" attributes.

2842 Implementations MAY, though they NEED NOT, include additional values in the  
2843 job's "job-state-reasons" attribute to indicate the progress of the job, such as  
2844 adding the 'job-printing' value to indicate when the output device is actually  
2845 making marks on paper and/or the 'processing-to-stop-point' value to indicate that  
2846 the IPP object is in the process of canceling or aborting the job. Most  
2847 implementations won't bother with this nuance.

2848  
2849 '6' 'processing-stopped': The job has stopped while processing for any number of reasons and  
2850 will return to the 'processing' state as soon as the reasons are no longer present.

2851

2852 The job's "job-state-reason" attribute MAY indicate why the job has stopped  
2853 processing. For example, if the output device is stopped, the 'printer-stopped'  
2854 value MAY be included in the job's "job-state-reasons" attribute.

2855

2856

2857

2858

2859

Note: When an output device is stopped, the device usually indicates its condition in human readable form locally at the device. A client can obtain more complete device status remotely by querying the Printer object's "printer-state", "printer-state-reasons" and "printer-state-message" attributes.

2860

2861

'7'

'canceled': The job has been canceled by a Cancel-Job operation and the Printer object has completed canceling the job and all job status attributes have reached their final values for the job. While the Printer object is canceling the job, the job remains in its current state, but the job's "job-state-reasons" attribute SHOULD contain the 'processing-to-stop-point' value and one of the 'canceled-by-user', 'canceled-by-operator', or 'canceled-at-device' value. When the job moves to the 'canceled' state, the 'processing-to-stop-point' value, if present, SHALL be removed, but the 'canceled-by-xxx', if present, SHALL remain.

2862

2863

2864

2865

2866

2867

2868

2869

2870

'8'

'aborted': The job has been aborted by the system, usually while the job was in the 'processing' or 'processing-stopped' state and the Printer has completed aborting the job and all job status attributes have reached their final values for the job. While the Printer object is aborting the job, the job remains in its current state, but the job's "job-state-reasons" attribute SHOULD contain the 'processing-to-stop-point' and 'aborted-by-system' values. When the job moves to the 'aborted' state, the 'processing-to-stop-point' value, if present, SHALL be removed, but the 'aborted-by-system' value, if present, SHALL remain.

2871

2872

2873

2874

2875

2876

2877

2878

2879

'9'

'completed': The job has completed successfully or with warnings or errors after processing and all of the job media sheets have been successfully stacked in the appropriate output bin(s) and all job status attributes have reached their final values for the job. The job's "job-state-reasons" attribute SHOULD contain one of: 'completed-successfully', 'completed-with-warnings', or 'completed-with-errors' values.

2880

2881

2882

2883

2884

2885

2886

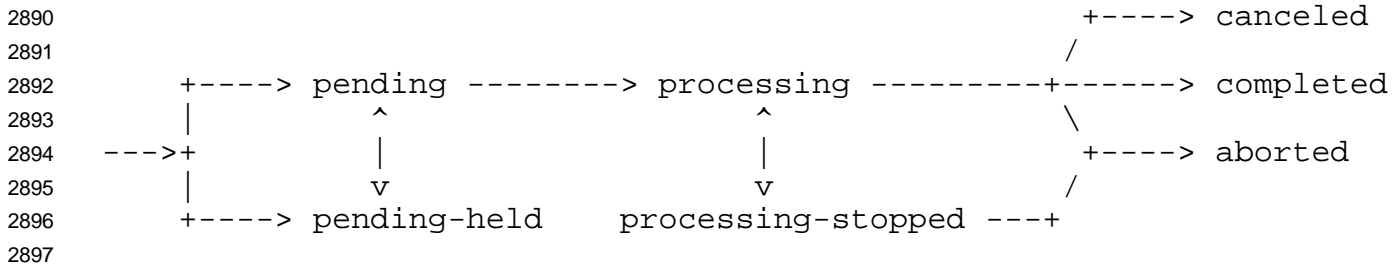
2887

2888

The final value for this attribute SHALL be one of: 'completed', 'canceled', or 'aborted' before the Printer removes the job altogether. The length of time that jobs remain in the 'canceled', 'aborted', and 'completed' states depends on implementation.

2889

The following figure shows the normal job state transitions.



2898 Normally a job progresses from left to right. Other state transitions are unlikely, but are not forbidden.  
 2899 Not shown are the transitions to the 'canceled' state from the 'pending', 'pending-held', and 'processing-  
 2900 stopped' states.

2901 Jobs reach one of the three terminal states: 'completed', 'canceled', or 'aborted', after the jobs have  
 2902 completed all activity, including stacking output media, after the jobs have completed all activity, and all  
 2903 job status attributes have reached their final values for the job.

2904 Note: As with all other IPP attributes, if the implementation can not determine the correct value for this  
 2905 attribute, it may choose to respond with the out-of-band value 'unknown' rather than try to guess at some  
 2906 possibly incorrect value and give the end user the wrong impression about the state of the Job object.  
 2907 For example, if the implementation is just a gateway into some printing system that does not provide  
 2908 detailed status about the print job, the IPP Job object's state might literally be 'unknown'.

#### 2909 4.3.8 job-state-reasons (1setOf type2 keyword)

2910 This attribute provides additional information about the job's current state, i.e., information that augments  
 2911 the value of the job's "job-state" attribute.

2912 Implementation of these values is OPTIONAL, i.e., a Printer NEED NOT implement them, even if (1)  
 2913 the output device supports the functionality represented by the reason and (2) is available to the Printer  
 2914 object implementation. These values MAY be used with any job state or states for which the reason  
 2915 makes sense. Furthermore, when implemented, the Printer SHALL return these values when the reason  
 2916 applies and SHALL NOT return them when the reason no longer applies whether the value of the Job's  
 2917 "job-state" attribute changed or not. When the Job does not have any reasons for being in its current  
 2918 state, the value of the Job's "job-state-reasons" attribute SHALL be 'none'.

2919 Note: While values cannot be added to the 'job-state' attribute without impacting deployed clients that  
 2920 take actions upon receiving "job-state" values, it is the intent that additional "job-state-reasons" values  
 2921 can be defined and registered without impacting such deployed clients. In other words, the "job-state-  
 2922 reasons" attribute is intended to be extensible.

2923 The following standard values are defined. For ease of understanding, the values are presented in the  
 2924 order in which the reasons are likely to occur (if implemented), starting with the 'job-incoming' value:

2925 'none': There are no reasons for the job's current state.  
2926 'job-incoming': The CreateJob operation has been accepted by the Printer, but the Printer is expecting  
2927 additional Send-Document and/or Send-URI operations and/or is accessing/accepting document  
2928 data.  
2929 'submission-interrupted': The job was not completely submitted for some unforeseen reason, such as:  
2930 (1) the Printer has crashed before the job was closed by the client, (2) the Printer or the document  
2931 transfer method has crashed in some non-recoverable way before the document data was entirely  
2932 transferred to the Printer, (3) the client crashed or failed to close the job before the time-out  
2933 period.  
2934 'job-outgoing': The Printer is transmitting the job to the output device.  
2935 'job-hold-until-specified': The value of the job's "job-hold-until" attribute was specified with a time  
2936 period that is still in the future. The job SHALL NOT be a candidate for processing until this  
2937 reason is removed and there are no other reasons to hold the job.  
2938 'resources-are-not-ready': At least one of the resources needed by the job, such as media, fonts,  
2939 resource objects, etc., is not ready on any of the physical printer's for which the job is a candidate.  
2940 This condition MAY be detected when the job is accepted, or subsequently while the job is  
2941 pending or processing, depending on implementation. The job may remain in its current state or  
2942 be moved to the 'pending-held' state, depending on implementation and/or job scheduling policy.  
2943 'printer-stopped-partly': The value of the Printer's "printer-state-reasons" attribute contains the value  
2944 'stopped-partly'.  
2945 'printer-stopped': The value of the Printer's "printer-state" attribute is 'stopped'.  
2946 'job-interpreting': Job is in the 'processing' state, but more specifically, the Printer is interpreting the  
2947 document data.  
2948 'job-queued': Job is in the 'processing' state, but more specifically, the Printer has queued the  
2949 document data.  
2950 'job-transforming': Job is in the 'processing' state, but more specifically, the Printer is interpreting  
2951 document data and producing another electronic representation.  
2952 'job-printing': The output device is marking media. This value is useful for Printers which spend a  
2953 great deal of time processing (1) when no marking is happening and then want to show that  
2954 marking is now happening or (2) when the job is in the process of being canceled or aborted while  
2955 the job remains in the 'processing' state, but the marking has not yet stopped so that impression or  
2956 sheet counts are still increasing for the job.  
2957 'job-canceled-by-user': The job was canceled by the owner of the job using the Cancel-Job request,  
2958 i.e., by a user whose authenticated identity is the same as the value of the originating user that  
2959 created the Job object, or by some other authorized end-user, such as a member of the job owner's  
2960 security group.  
2961 'job-canceled-by-operator': The job was canceled by the operator using the Cancel-Job request, i.e.,  
2962 by a user who has been authenticated as having operator privileges (whether local or remote). If  
2963 the security policy is to allow anyone to cancel anyone's job, then this value may be used when the

2964 job is canceled by other than the owner of the job. For such a security policy, in effect, everyone  
2965 is an operator as far as canceling jobs with IPP is concerned.

2966 'job-canceled-at-device': The job was canceled by an unidentified local user, i.e., a user at a console  
2967 at the device.

2968 'aborted-by-system': The job (1) is in the process of being aborted, (2) has been aborted by the  
2969 system and placed in the 'aborted' state, or (3) has been aborted by the system and placed in the  
2970 'pending-held' state, so that a user or operator can manually try the job again.

2971 'processing-to-stop-point': The requester has issued a Cancel-job operation or the Printer object has  
2972 aborted the job, but is still performing some actions on the job until a specified stop point occurs  
2973 or job termination/cleanup is completed.

2974

2975 This reason is recommended to be used in conjunction with the 'processing' job state to indicate  
2976 that the Printer object is still performing some actions on the job while the job remains in the  
2977 'processing' state. After all the job's job description attributes have stopped incrementing, the  
2978 Printer object moves the job from the 'processing' state to the 'canceled' or 'aborted' job states.

2979

2980 'service-off-line': The Printer is off-line and accepting no jobs. All 'pending' jobs are put into the  
2981 'pending-held' state. This situation could be true if the service's or document transform's input is  
2982 impaired or broken.

2983 'job-completed-successfully': The job completed successfully.

2984 'job-completed-with-warnings': The job completed with warnings.

2985 'job-completed-with-errors': The job completed with errors (and possibly warnings too).

2986

#### 2987 4.3.9 job-state-message (text(MAX))

2988 This attribute specifies information about the "job-state" and "job-state-reasons" attributes in human  
2989 readable text. If the Printer object supports this attribute, the Printer object SHALL be able to generate  
2990 this message in any of the natural languages identified by the Printer's "generated-natural-language-  
2991 supported" attribute (see the "attributes-natural-language" operation attribute specified in Section  
2992 3.1.4.1).

2993 Note: the value SHOULD NOT contain additional information not contained in the values of the "job-  
2994 state" and "job-states-reasons" attributes, such as interpreter error information. Otherwise, application  
2995 programs might attempt to parse the (localized text). For such additional information such as interpreter  
2996 errors for application program consumption, a new attribute with keyword values, needs to be developed  
2997 and registered.

## 2998 4.3.10 number-of-documents (integer(0:MAX))

2999 This attribute indicates the number of documents in the job, i.e., the number of Send-Document, Send-  
3000 URI, Print-Job, or Print-URI operations that the Printer has accepted for this job, regardless of whether  
3001 the document data has reached the Printer object or not.

3002 Implementations supporting the OPTIONAL Create-Job/Send-Document/Send-URI operations  
3003 SHOULD support this attribute so that clients can query the number of documents in each job.

## 3004 4.3.11 output-device-assigned (name(127))

3005 This attribute identifies the output device to which the Printer object has assigned this job. If an output  
3006 device implements an embedded Printer object, the Printer object NEED NOT set this attribute. If a print  
3007 server implements a Printer object, the value MAY be empty (zero-length string) or not returned until the  
3008 Printer object assigns an output device to the job. This attribute is particularly useful when a single  
3009 Printer object support multiple devices (so called "fan-out").

## 3010 4.3.12 time-at-creation (integer(0:MAX))

3011 This attribute indicates the point in time at which the Job object was created. In order to populate this  
3012 attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object is  
3013 created.

## 3014 4.3.13 time-at-processing (integer(0:MAX))

3015 This attribute indicates the point in time at which the Job object began processing. In order to populate  
3016 this attribute, the Printer object uses the value in its "printer-up-time" attribute at the time the Job object  
3017 is moved into the 'processing' state for the first time.

## 3018 4.3.14 time-at-completed (integer(0:MAX))

3019 This attribute indicates the point in time at which the Job object completed (or was cancelled or aborted).  
3020 In order to populate this attribute, the Printer object uses the value in its "printer-up-time" attribute at the  
3021 time the Job object is moved into the 'completed' or 'canceled' or 'aborted' state.

## 3022 4.3.15 number-of-intervening-jobs (integer(0:MAX))

3023 This attribute indicates the number of jobs that are "ahead" of this job in the relative chronological order  
3024 of expected time to complete (i.e., the current scheduled order). For efficiency, it is only necessary to  
3025 calculate this value when an operation is performed that requests this attribute.

## 3026 4.3.16 job-message-from-operator (text(127))

3027 This attribute provides a message from an operator, system administrator or "intelligent" process to  
3028 indicate to the end user the reasons for modification or other management action taken on a job.

## 3029 4.3.17 job-k-octets (integer(0:MAX))

3030 This attribute specifies the total size of the document(s) in K octets, i.e., in units of 1024 octets requested  
3031 to be processed in the job. The value SHALL be rounded up, so that a job between 1 and 1024 octets  
3032 SHALL be indicated as being 1, 1025 to 2048 SHALL be 2, etc.

3033 This value SHALL ~~not~~**NOT** include the multiplicative factors contributed by the number of copies  
3034 specified by the "copies" attribute, independent of whether the device can process multiple copies without  
3035 making multiple passes over the job or document data and independent of whether the output is collated  
3036 or not. Thus the value is independent of the implementation and indicates the size of the document(s)  
3037 measured in K octets independent of the number of copies.

3038 This value SHALL also not include the multiplicative factor due to a copies instruction embedded in the  
3039 document data. If the document data actually includes replications of the document data, this value will  
3040 include such replication. In other words, this value is always the size of the source document data, rather  
3041 than a measure of the hardcopy output to be produced.

3042 Note: This attribute and the following two attributes ("job-impressions" and "job-media-sheets") are not  
3043 intended to be counters; they are intended to be useful routing and scheduling information if known. For  
3044 these three attributes, the Printer object may try to compute the value if it is not supplied in the create  
3045 request. Even if the client does supply a value for these three attributes in the create request, the Printer  
3046 object MAY choose to change the value if the Printer object is able to compute a value which is more  
3047 accurate than the client supplied value. The Printer object may be able to determine the correct value for  
3048 these three attributes either right at job submission time or at any later point in time.

## 3049 4.3.18 job-impressions (integer(0:MAX))

3050 This attribute specifies the total size in number of impressions of the document(s) being submitted (see  
3051 the definition of impression in section 13.2.5).

3052 As with "job-k-octets", this value SHALL ~~not~~**NOT** include the multiplicative factors contributed by the  
3053 number of copies specified by the "copies" attribute, independent of whether the device can process  
3054 multiple copies without making multiple passes over the job or document data and independent of  
3055 whether the output is collated or not. Thus the value is independent of the implementation and reflects  
3056 the size of the document(s) measured in impressions independent of the number of copies.



3057 As with "job-k-octets", this value SHALL also not include the multiplicative factor due to a copies  
3058 instruction embedded in the document data. If the document data actually includes replications of the  
3059 document data, this value will include such replication. In other words, this value is always the number of  
3060 impressions in the source document data, rather than a measure of the number of impressions to be  
3061 produced by the job.

3062 See the Note in the "job-k-octets" attribute that also applies to this attribute.

#### 3063 4.3.19 job-media-sheets (integer(0:MAX))

3064 This attribute specifies the total number of media sheets to be produced for this job.

3065 Unlike the "job-k-octets" and the "job-impressions" attributes, this value SHALL include the  
3066 multiplicative factors contributed by the number of copies specified by the "copies" attribute and a  
3067 'number of copies' instruction embedded in the document data, if any. This difference allows the system  
3068 administrator to control the lower and upper bounds of both (1) the size of the document(s) with "job-k-  
3069 octets-supported" and "job-impressions-supported" and (2) the size of the job with "job-media-sheets-  
3070 supported".

3071 See the Note in the "job-k-octets" attribute that also applies to this attribute.

#### 3072 4.3.20 job-k-octets-processed (integer(0:MAX))

3073 This attribute specifies the total number of octets processed in K octets, i.e., in units of 1024 octets so  
3074 far. The value SHALL be rounded up, so that a job between 1 and 1024 octets inclusive SHALL be  
3075 indicated as being 1, 1025 to 2048 inclusive SHALL be 2, etc.

3076 For implementations where multiple copies are produced by the interpreter with only a single pass over  
3077 the data, the final value SHALL be equal to the value of the "job-k-octets" attribute. For  
3078 implementations where multiple copies are produced by the interpreter by processing the data for each  
3079 copy, the final value SHALL be a multiple of the value of the "job-k-octets" attribute.

3080 Note: This attribute and the following two attributes ("job-impressions-completed" and "job-sheets-  
3081 completed") are intended to be counters. That is, the value for a job that has not started processing  
3082 SHALL be 0. When the job's "job-state" is 'processing' or 'processing-stopped', this value is intended to  
3083 contain the amount of the job that has been processed to the time at which the attributes are requested.

#### 3084 4.3.21 job-impressions-completed (integer(0:MAX))

3085 This job attribute specifies the number of impressions completed for the job so far. For printing devices,  
3086 the impressions completed includes interpreting, marking, and stacking the output.

3087 See the note in "job-k-octets-processed" which also applies to this attribute.

#### 3088 4.3.22 job-media-sheets-completed (integer(0:MAX))

3089 This job attribute specifies the media-sheets completed marking and stacking for the entire job so far  
3090 whether those sheets have been processed on one side or on both.

3091 See the note in "job-k-octets-processed" which also applies to this attribute.

#### 3092 4.3.23 attributes-charset (charset)

3093 This MANDATORY attribute is populated using the value in the client supplied "attributes-charset"  
3094 attribute in the create request. It identifies the charset (coded character set and encoding method) used  
3095 by any Job attributes with attribute syntax 'text' and 'name' that were supplied by the client in the create  
3096 request. See Section 3.1.4 for a complete description of the "attributes-charset" operation attribute.

3097 This attribute does not indicate the charset in which the 'text' and 'name' values are stored internally in the  
3098 Job object. The internal charset is implementation-defined. The IPP object SHALL convert from  
3099 whatever the internal charset is to that being requested in an operation as specified in Section 3.1.4.

#### 3100 4.3.24 attributes-natural-language (naturalLanguage)

3101 This MANDATORY attribute is populated using the value in the client supplied "attributes-natural-  
3102 language" attribute in the create request. It identifies the natural language used for any Job attributes  
3103 with attribute syntax 'text' and 'name' that were supplied by the client in the create request. See Section  
3104 3.1.4 for a complete description of the "attributes-natural-language" operation attribute. See Section  
3105 3.2.6 for how this attribute is returned in a Get-Jobs operation when jobs with different natural languages  
3106 are returned. See Sections 4.1.1.2 and 4.1.2.2 for how a Natural Language Override may be supplied  
3107 explicitly for each 'text' and 'name' attribute value that differs from the value identified by the "attributes-  
3108 natural-language" attribute.

#### 3109 4.4 Printer Description Attributes

3110 These attributes form the attribute group called "printer-description". The following table summarizes  
3111 these attributes, their syntax, and whether or not they are MANDATORY for a Printer object to support.  
3112 If they are not indicated as MANDATORY, they are OPTIONAL. The maximum size in octets for 'text'  
3113 and 'name' attributes is indicated in parentheses.

3114 Note: How these attributes are set by an Administrator is outside the scope of this specification.

3115	+-----+-----+-----+
3116	Attribute   Syntax   MANDATORY?
3117	+-----+-----+-----+
3118	printer-uri-supported   lsetOf uri   MANDATORY
3119	+-----+-----+-----+
3120	uri-security-supported   lsetOf type2 keyword   MANDATORY
3121	+-----+-----+-----+
3122	printer-name   name (127)   MANDATORY
3123	+-----+-----+-----+
3124	printer-location   text (127)
3125	+-----+-----+-----+
3126	printer-info   text (127)
3127	+-----+-----+-----+
3128	printer-more-info   uri
3129	+-----+-----+-----+
3130	printer-driver-installer   uri
3131	+-----+-----+-----+
3132	printer-make-and-model   text (127)
3133	+-----+-----+-----+
3134	printer-more-info-
3135	manufacturer   uri
3136	+-----+-----+-----+
3137	printer-state   type1 enum   MANDATORY
3138	+-----+-----+-----+
3139	printer-state-reasons   lsetOf type2 keyword
3140	+-----+-----+-----+
3141	printer-state-message   text (MAX)
3142	+-----+-----+-----+
3143	operations-supported   lsetOf type2 enum   MANDATORY
3144	+-----+-----+-----+
3145	charset-configured   charset   MANDATORY
3146	+-----+-----+-----+
3147	charset-supported   lsetOf charset   MANDATORY
3148	+-----+-----+-----+
3149	natural-language-configured   naturalLanguage   MANDATORY
3150	+-----+-----+-----+
3151	generated-natural-language-
3152	supported   lsetOf   MANDATORY
3153	+-----+-----+-----+
3154	document-format-default   mimeType   MANDATORY
3155	+-----+-----+-----+
3156	document-format-
3157	supported   lsetOf   MANDATORY
3158	+-----+-----+-----+
3159	printer-is-accepting-jobs   boolean   MANDATORY

3160	+-----+-----+-----+
3161	queued-job-count   integer (0:MAX)
3162	+-----+-----+-----+
3163	printer-message-from-   text (127)
3164	operator
3165	+-----+-----+-----+
3166	color-supported   boolean
3167	+-----+-----+-----+
3168	reference-uri-schemes-   lsetOf uriScheme
3169	supported
3170	+-----+-----+-----+
3171	pdl-override-supported   type2 keyword   MANDATORY
3172	+-----+-----+-----+
3173	printer-up-time   integer (1:MAX)   MANDATORY
3174	+-----+-----+-----+
3175	printer-current-time   dateTime
3176	+-----+-----+-----+
3177	multiple-operation-time-out   integer (1:MAX)
3178	+-----+-----+-----+
3179	compression-supported   lsetOf type3 keyword
3180	+-----+-----+-----+
3181	job-k-octets-supported   rangeOfInteger
3182	(0:MAX)
3183	+-----+-----+-----+
3184	job-impressions-supported   rangeOfInteger
3185	(0:MAX)
3186	+-----+-----+-----+
3187	job-media-sheets-supported   rangeOfInteger
3188	(0:MAX)
3189	+-----+-----+-----+
3190	

#### 3191 4.4.1 printer-uri-supported (lsetOf uri)

3192 This MANDATORY Printer attribute contains at least one URI for the Printer object. It OPTIONALLY  
3193 contains more than one URI for the Printer object. An administrator determines a Printer object's  
3194 URI(s) and configures this attribute to contain those URIs by some means outside the scope of IPP/1.0.  
3195 The precise format of this URI is implementation dependent and depends on the protocol. See the next  
3196 section for a description "uri-security-supported" which is the MANDATORY companion attribute to  
3197 this "printer-uri-supported" attribute. See section 2.4 on Printer object identity and section 8.2 on  
3198 security and URIs for more information.

## 3199 4.4.2 uri-security-supported (1setOf type2 keyword)

3200 This MANDATORY Printer attribute MUST have the same cardinality (contain the same number of  
 3201 values) as the "printer-uri-supported" attribute. This attribute identifies the security mechanisms used for  
 3202 each URI listed in the "printer-uri-supported" attribute. The "i th" value in "uri-security-supported"  
 3203 corresponds to the "i th" value in "printer-uri-supported" and it describes the security mechanisms used  
 3204 for accessing the Printer object via that URI. The following standard values are defined:

3205 'none': There are no secure communication channel protocols in use for the given URI.

3206 'tls': TLS 1.0 [TLS] is the secure communications channel protocol in use for the given URI.

3207 'ssl3': SSL3 is the secure communications channel protocol in use for the given URI.

3208

3209 Consider the following example. ~~For a single Printer object, An-an~~ administrator configures the "printer-  
 3210 uri-supported" and "uri-security-supported" attributes as follows:

3211 "printer-uri-supported": 'http://acme.com/open-use-printer', 'http://acme.com/~~passwordrestricted-use-~~  
 3212 printer', 'http://acme.com/private-printer', ~~'https://acme.com/private-printer'~~

3213 "uri-security-supported": 'none', 'none', 'tls', ~~'ssl3'~~

3214

3214

3215 In this case, one Printer object has ~~four~~ three URIs.

- 3216 - For the first URI, 'http://acme.com/open-use-printer', the value 'none' in "uri-security-supported"  
 3217 indicates that there is no secure channel protocol configured to run under HTTP. The name  
 3218 implies that there is no Basic or Digest authentication being used, but it is up to the client to  
 3219 determine that while using HTTP underneath the IPP application protocol.
- 3220 - For the second URI, 'http://acme.com/~~passwordrestricted-use-~~printer', the value 'none' in "uri-  
 3221 security-supported" indicates that there is no secure channel protocol configured to run under  
 3222 HTTP. ~~However, i~~In this case, although the name does imply that there is some sort of Basic or  
 3223 Digest authentication being used within HTTP, ~~But again,~~ it is up to the client to determine that  
 3224 while using HTTP and by processing any '401 Unauthorized' HTTP error messages.
- 3225 - For the third URI, 'http://acme.com/private-printer', the value 'tls' in "uri-security-supported"  
 3226 indicates that TLS is being used to secure the channel. The client SHOULD be prepared to use  
 3227 TLS framing to negotiate an acceptable ciphersuite to use while communicating with the Printer  
 3228 object. In this case, the name implies the use of a secure communications channel, but the fact is  
 3229 made explicit by the presence of the 'tls' value in "uri-security-supported". The client does not  
 3230 need to resort to understanding which security it must use by following naming conventions or by  
 3231 parsing the URI to determine which security mechanisms are implied.
- ~~3232 -For the fourth URI, 'https://acme.com/private-printer', the value 'ssl3' in "uri-security-supported"~~  
 3233 ~~indicates that SSL3 is being used to secure the channel. Notice that the URI is the same as the~~  
 3234 ~~URI in the third case (except for the scheme), however this time, the client SHOULD be prepared~~  
 3235 ~~to use SSL3 framing to negotiate an acceptable ciphersuite to use while communicating with the~~

3236 ~~Printer object. As in the third case, the presence of a secure channel and which security~~  
3237 ~~mechanism to use is made explicit by the presence of the 'ssl3' value in "uri-security-supported".~~

3238  
3239 It is expected that many IPP Printer objects will be configured to support only one channel (either  
3240 configured to use TLS access or not), and will therefore only ever have one URI listed in the "printer-uri-  
3241 supported" attribute. No matter the configuration of the Printer object (whether it has only one URI or  
3242 more than one URI), a client SHALL supply only one URI in the target "printer-uri" operation attribute.

#### 3243 4.4.3 printer-name (name(127))

3244 This MANDATORY Printer attribute contains the name of the Printer object. It is a name that is more  
3245 end-user friendly than a URI. An administrator determines a printer's name and sets this attribute to that  
3246 name. This name may be the last part of the printer's URI or it may be unrelated. In non-US-English  
3247 locales, a name may contain characters that are not allowed in a URI.

#### 3248 4.4.4 printer-location (text(127))

3249 This Printer attribute identifies the location of the device. This could include things like: "in Room 123A,  
3250 second floor of building XYZ".

#### 3251 4.4.5 printer-info (text(127))

3252 This Printer attribute identifies the descriptive information about this Printer object. This could include  
3253 things like: "This printer can be used for printing color transparencies for HR presentations", or "Out of  
3254 courtesy for others, please print only small (1-5 page) jobs at this printer", or even "This printer is going  
3255 away on July 1, 1997, please find a new printer".

#### 3256 4.4.6 printer-more-info (uri)

3257 This Printer attribute contains a URI used to obtain more information about this specific Printer object.  
3258 For example, this could be an HTTP type URI referencing an HTML page accessible to a Web Browser.  
3259 The information obtained from this URI is intended for end user consumption. Features outside the scope  
3260 of IPP can be accessed from this URI. The information is intended to be specific to this printer instance  
3261 and site specific services (e.g. job pricing, services offered, end user assistance). The device manufacturer  
3262 may initially populate this attribute.

## 3263 4.4.7 printer-driver-installer (uri)

3264 This Printer attribute contains a URI to use to locate the driver installer for this Printer object. This  
 3265 attribute is intended for consumption by automata. The mechanics of print driver installation is outside  
 3266 the scope of IPP. The device manufacturer may initially populate this attribute.

## 3267 4.4.8 printer-make-and-model (text(127))

3268 This Printer attribute identifies the make and model of the device. The device manufacturer may initially  
 3269 populate this attribute.

## 3270 4.4.9 printer-more-info-manufacturer (uri)

3271 This Printer attribute contains a URI used to obtain more information about this type of device. The  
 3272 information obtained from this URI is intended for end user consumption. Features outside the scope of  
 3273 IPP can be accessed from this URI (e.g., latest firmware, upgrades, print drivers, optional features  
 3274 available, details on color support). The information is intended to be germane to this printer without  
 3275 regard to site specific modifications or services. The device manufacturer may initially populate this  
 3276 attribute.

## 3277 4.4.10 printer-state (type1 enum)

3278 This MANDATORY Printer attribute identifies the current state of the device. The "printer-state  
 3279 reasons" attribute augments the "printer-state" attribute to give more detailed information about the  
 3280 Printer in the given printer state.

3281 A Printer object need only update this attribute before responding to an operation which requests the  
 3282 attribute; the Printer object NEED NOT update this attribute continually, since asynchronous event  
 3283 notification is not part of IPP/1.0. A Printer NEED NOT implement all values if they are not applicable  
 3284 to a given implementation.

3285 The following standard values are defined:

3286	Value	Symbolic Name and Description
3287		
3288	'3'	'idle': If a Printer receives a job (whose required resources are ready) while in this state, 3289 such a job SHALL transit into the processing state immediately. If the printer- 3290 state-reasons attribute contains any reasons, they SHALL be reasons that would 3291 not prevent a job from transiting into the processing state immediately, e.g., toner- 3292 low. Note: if a Printer controls more than one output device, the above definition 3293 implies that a Printer is idle if at least one output device is idle.

3294  
3295 '4' 'processing': If a Printer receives a job (whose required resources are ready) while in this  
3296 state, such a job SHALL transit into the pending state immediately. Such a job  
3297 SHALL transit into the processing state only after jobs ahead of it complete. If the  
3298 printer-state-reasons attribute contains any reasons, they SHALL be reasons that  
3299 do not prevent the current job from printing, e.g. toner-low. Note: if a Printer  
3300 controls more than one output device, the above definition implies that a Printer is  
3301 processing if at least one output device is processing, and none is idle.

3302  
3303 '5' 'stopped': If a Printer receives a job (whose required resources are ready) while in this  
3304 state, such a job SHALL transit into the pending state immediately. Such a job  
3305 SHALL transit into the processing state only after some human fixes the problem  
3306 that stopped the printer and after jobs ahead of it complete printing. If supported,  
3307 the "printer-state-reasons" attribute SHALL contain at least one reason, e.g.  
3308 media-jam, which prevents it from either processing the current job or transitioning  
3309 a pending job to the processing state.

3310  
3311 Note: if a Printer controls more than one output device, the above definition  
3312 implies that a Printer is stopped only if all output devices are stopped. Also, it is  
3313 tempting to define stopped as when a sufficient number of output devices are  
3314 stopped and leave it to an implementation to define the sufficient number. But  
3315 such a rule complicates the definition of stopped and processing. For example,  
3316 with this alternate definition of stopped, a job can move from idle to processing  
3317 without human intervention, even though the Printer is stopped.

3318

#### 3319 4.4.11 printer-state-reasons (1setOf type2 keyword)

3320 This Printer attribute supplies additional detail about the device's state.

3321 Each keyword value MAY have a suffix to indicate its level of severity. The three levels are: report (least  
3322 severe), warning, and error (most severe).

- 3323 - '-report': This suffix indicates that the reason is a "report". An implementation may choose to omit  
3324 some or all reports. Some reports specify finer granularity about the printer state; others serve as  
3325 a precursor to a warning. A report SHALL contain nothing that could affect the printed output.  
3326 - '-warning': This suffix indicates that the reason is a "warning". An implementation may choose to  
3327 omit some or all warnings. Warnings serve as a precursor to an error. A warning SHALL contain  
3328 nothing that prevents a job from completing, though in some cases the output may be of lower  
3329 quality.



3330 - '-error': This suffix indicates that the reason is an "error". An implementation SHALL include all  
3331 errors. If this attribute contains one or more errors, printer SHALL be in the stopped state.

3332  
3333 If the implementation does not add any one of the three suffixes, all parties SHALL assume that the  
3334 reason is an "error".

3335 If a Printer object controls more than one output device, each value of this attribute MAY apply to one or  
3336 more of the output devices. An error on one output device that does not stop the Printer object as a  
3337 whole MAY appear as a warning in the Printer's "printer-state-reasons attribute". If the "printer-state"  
3338 for such a Printer has a value of 'stopped', then there MUST be an error reason among the values in the  
3339 "printer-state-reasons" attribute.

3340 The following standard values are defined:

3341 'other': The device has detected an error other than one listed in this document.

3342 'none': There are not reasons. This state reason is semantically equivalent to "printer-state-reasons"  
3343 without any value.

3344 'media-needed': A tray has run out of media.

3345 'media-jam': The device has a media jam.

3346 'paused': Someone has paused the Printer object. In this state, a Printer SHALL ~~not~~ NOT produce  
3347 printed output, but it SHALL perform other operations requested by a client. If a Printer had  
3348 been printing a job when the Printer was paused, the Printer SHALL resume printing that job  
3349 when the Printer is no longer paused and leave no evidence in the printed output of such a pause.

3350 'shutdown': Someone has removed a Printer object from service, and the device may be powered  
3351 down or physically removed. In this state, a Printer object SHALL ~~not~~ NOT produce printed  
3352 output, and unless the Printer object is realized by a print server that is still active, the Printer  
3353 object SHALL perform no other operations requested by a client, including returning this value. If  
3354 a Printer object had been printing a job when it was shutdown, the Printer need not resume  
3355 printing that job when the Printer is no longer shutdown. If the Printer resumes printing such a  
3356 job, it may leave evidence in the printed output of such a shutdown, e.g. the part printed before  
3357 the shutdown may be printed a second time after the shutdown.

3358 'connecting-to-device': The Printer object has scheduled a job on the output device and is in the  
3359 process of connecting to a shared network output device (and might not be able to actually start  
3360 printing the job for an arbitrarily long time depending on the usage of the output device by other  
3361 servers on the network).

3362 'timed-out': The server was able to connect to the output device (or is always connected), but was  
3363 unable to get a response from the output device.

3364 'stopping': The Printer object is in the process of stopping the device and will be stopped in a while.  
3365 When the device is stopped, the Printer object will change the Printer object's state to 'stopped'.  
3366 The 'stopping-warning' reason is never an error, even for a Printer with a single output device.

3367           When an output-device ceases accepting jobs, the Printer will have this reason while the output  
3368           device completes printing.

3369           'stopped-partly': When a Printer object controls more than one output device, this reason indicates  
3370           that one or more output devices are stopped. If the reason is a report, fewer than half of the  
3371           output devices are stopped. If the reason is a warning, fewer than all of the output devices are  
3372           stopped.

3373           'toner-low': The device is low on toner.

3374           'marker-supply-low': The device is low on marker supply (ink, paint, etc.).

3375           'spool-area-full': The limit of persistent storage allocated for spooling has been reached.

3376           'cover-open': One or more covers on the device are open.

3377           'interlock-open': One or more interlock devices on the printer are unlocked.

3378           'door-open': One or more doors on the device are open.

3379           'input-tray-missing': One or more input trays are not in the device.

3380           'media-low': At least one input tray is low on media.

3381           'media-empty': At least one input tray is empty.

3382           'output-tray-missing': One or more output trays are not in the device

3383           'output-area-almost-full': One or more output area is almost full (e.g. tray, stacker, collator).

3384           'output-area-full': One or more output area is full. (e.g. tray, stacker, collator)

3385           'marker-supply-low': The device is low on at least one marker supply. (e.g. toner, ink, ribbon)

3386           'marker-supply-empty': The device is out of at least one marker supply. (e.g. toner, ink, ribbon)

3387           'marker-waste-almost-full': The device marker supply waste receptacle is almost full.

3388           'marker-waste-full': The device marker supply waste receptacle is full.

3389           'fuser-over-temp': The fuser temperature is above normal.

3390           'fuser-under-temp': The fuser temperature is below normal.

3391           'opc-near-eol': The optical photo conductor is near end of life.

3392           'opc-life-over': The optical photo conductor is no longer functioning.

3393           'developer-low': The device is low on developer.

3394           'developer-empty': The device is out of developer.

3395           'interpreter-resource-unavailable': An interpreter resource is unavailable (i.e. font, form)

3396

#### 3397 4.4.12 printer-state-message (text(MAX))

3398           This Printer attribute specifies the additional information about the printer state and printer state reasons  
3399           in human readable text. If the Printer object supports this attribute, the Printer object SHALL be able to  
3400           generate this message in any of the natural languages identified by the Printer's "generated-natural-  
3401           language-supported" attribute (see the "attributes-natural-language" operation attribute specified in  
3402           Section 3.1.4.1).

## 3403 4.4.13 operations-supported (1setOf type2 enum)

3404 This MANDATORY Printer attribute specifies the set of supported operations for this Printer object and  
 3405 contained Job objects. No 32-bit enum value for this attribute SHALL exceed 0x8FFF, since these values  
 3406 are passed in two octets in each Protocol request [IPP-PRO].

3407 The following standard values are defined:

3408	Value	Operation Name
3409	-----	-----
3410		
3411	0x0000	reserved, not used
3412	0x0001	reserved, not used
3413	0x0002	Print-Job
3414	0x0003	Print-URI
3415	0x0004	Validate-Job
3416	0x0005	Create-Job
3417	0x0006	Send-Document
3418	0x0007	Send-URI
3419	0x0008	Cancel-Job
3420	0x0009	Get-Job-Attributes
3421	0x000A	Get-Jobs
3422	0x000B	Get-Printer-Attributes
3423	0x000C-0x3FFF	reserved for future operations
3424	0x4000-0x8FFF	reserved for private extensions

3425

3426 This allows for certain vendors to implement private extensions that are guaranteed to not conflict with  
 3427 future registered extensions. However, there is no guarantee that two or more private extensions will not  
 3428 conflict.

## 3429 4.4.14 charset-configured (charset)

3430 This MANDATORY Printer attribute identifies the charset that the Printer object has been configured to  
 3431 represent 'text' and 'name' Printer attributes that are set by the operator, system administrator, or  
 3432 manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info" (text), and "printer-  
 3433 make-and-model" (text). Therefore, the value of the Printer object's "charset-configured" attribute  
 3434 SHALL also be among the values of the Printer object's "charset-supported" attribute.

## 3435 4.4.15 charset-supported (1setOf charset)

3436 This MANDATORY Printer attribute identifies the set of charsets that the Printer and contained Job  
3437 objects support in attributes with attribute syntax 'text' and 'name'. At least the value 'utf-8' SHALL be  
3438 present, since IPP objects MUST support the UTF-8 [RFC2044] charset. If a Printer object supports a  
3439 charset, it means that for all attributes of syntax 'text' and 'name' the IPP object SHALL (1) accept the  
3440 charset in requests and return the charset in responses as needed.

3441 If more charsets than UTF-8 are supported, the IPP object SHALL perform charset conversion between  
3442 the charsets as described in Section 3.2.1.2.

## 3443 4.4.16 natural-language-configured (naturalLanguage)

3444 This MANDATORY Printer attribute identifies the natural language that the Printer object has been  
3445 configured to represent 'text' and 'name' Printer attributes that are set by the operator, system  
3446 administrator, or manufacturer, i.e., for "printer-name" (name), "printer-location" (text), "printer-info"  
3447 (text), and "printer-make-and-model" (text). When returning these Printer attributes, the Printer object  
3448 MAY return them in the configured natural language specified by this attribute, instead of the natural  
3449 language requested by the client in the "attributes-natural-language" operation attribute. See Section  
3450 3.1.4.1 for the specification of the OPTIONAL multiple natural language support. Therefore, the value  
3451 of the Printer object's "natural-language-configured" attribute SHALL also be among the values of the  
3452 Printer object's "generated-natural-language-supported" attribute.

## 3453 4.4.17 generated-natural-language-supported (1setOf naturalLanguage)

3454 This MANDATORY Printer attribute identifies the natural language(s) that the Printer object and  
3455 contained Job objects support in attributes with attribute syntax 'text' and 'name'. The natural language(s)  
3456 supported depends on implementation and/or configuration. Unlike charsets, IPP objects SHALL accept  
3457 requests with any natural language or any Natural Language Override whether the natural language is  
3458 supported or not.

3459 If a Printer object supports a natural language, it means that for any of the attributes for which the Printer  
3460 or Job object generates messages, i.e., for the "job-state-message" and "printer-state-message" attributes  
3461 and Operation Messages (see Section 3.1.5) in operation responses, the Printer and Job objects SHALL  
3462 be able to generate messages in any of the Printer's supported natural languages. See section 3.1.4 for  
3463 the specification of 'text' and 'name' attributes in operation requests and responses.

3464 Note: A Printer object that supports multiple natural languages, often has separate catalogs of messages,  
3465 one for each natural language supported.

3466 4.4.18 document-format-default (mimeMediaType)

3467 This Printer attribute identifies the document format that the Printer object has been configured to assume  
3468 if the client does not supply a "document-format" operation attribute in any of the operation requests that  
3469 supply document data. The standard values for this attribute are Internet Media types (sometimes called  
3470 MIME types). For further details see the description of the 'mimeMediaType' attribute syntax in Section  
3471 4.1.9.

3472 4.4.19 document-format-supported (1setOf mimeMediaType)

3473 This Printer attribute identifies the set of document formats that the Printer object and contained Job  
3474 objects can support. For further details see the description of the 'mimeMediaType' attribute syntax in  
3475 Section 4.1.9.

3476 4.4.20 printer-is-accepting-jobs (boolean)

3477 This MANDATORY Printer attribute indicates whether the printer is currently able to accept jobs, i.e., is  
3478 accepting Print-Job, Print-URI, and Create-Job requests. If the value is 'true', the printer is accepting  
3479 jobs. If the value is 'false', the Printer object is currently rejecting any jobs submitted to it. In this case,  
3480 the Printer object returns the 'server-error-not-accepting-jobs' status code.

3481 Note: This value is independent of the "printer-state" and "printer-state-reasons" attributes because its  
3482 value does not affect the current job; rather it affects future jobs. This attribute may cause the Printer to  
3483 reject jobs when the "printer-state" is 'idle' or it may cause the Printer object to accept jobs when the  
3484 "printer-state" is 'stopped'.

3485 4.4.21 queued-job-count (integer(0:MAX))

3486 This Printer attribute contains a count of the number of jobs that are either 'pending', 'processing',  
3487 'pending-held', or 'processing-stopped' and is set by the Printer object.

3488 4.4.22 printer-message-from-operator (text(127))

3489 This Printer attribute provides a message from an operator, system administrator or "intelligent" process  
3490 to indicate to the end user information or status of the printer, such as why it is unavailable or when it is  
3491 expected to be available.

## 3492 4.4.23 color-supported (boolean)

3493 This Printer attribute identifies whether the device is capable of any type of color printing at all, including  
3494 highlight color. All document instructions having to do with color are embedded within the document  
3495 PDL (none are external IPP attributes in IPP/1.0).

3496 Note: end-users are able to determine the nature and details of the color support by querying the  
3497 "printer-more-info-manufacturer" Printer attribute.

## 3498 4.4.24 reference-uri-schemes-supported (1setOf uriScheme)

3499 This Printer attribute specifies which URI schemes are supported for use in the "document-uri" operation  
3500 attribute of the Print-URI or Send-URI operation. If a Printer object supports these optional operations,  
3501 it MUST support the "reference-uri-schemes-supported" Printer attribute with at least the following  
3502 schemed URI values:

3503 'ftp': The Printer object will use an FTP 'get' operation as defined in RFC 1738 and RFC 2316.~~If~~  
3504 ~~the URI does not indicate a name or password in the URI itself, the Printer object will use~~  
3505 ~~anonymous FTP generating (if prompted) a password. Since many FTP servers require that~~  
3506 ~~anonymous FTP logins supply a password in the form a valid Internet email address, the Printer~~  
3507 ~~object MUST be able to generate such a password (syntactically correct, yet perhaps semantically~~  
3508 ~~meaningless) if needed.~~

3509  
3510 The Printer object MAY OPTIONALLY support other URI schemes (see section 4.1.6).

## 3511 4.4.25 pdl-override-supported (type2 keyword)

3512 This MANDATORY Printer attribute expresses the ability for a particular Printer implementation to  
3513 either attempt to override document data instructions with IPP attributes or not.

3514 This attribute takes on the following values:

- 3515 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values  
3516 take precedence over embedded instructions in the document data, however there is no guarantee.
- 3517 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP  
3518 attribute values take precedence over embedded instructions in the document data.

3519  
3520 Section 16 contains a full description of how this attribute interacts with and affects other IPP attributes,  
3521 especially the "ipp-attribute-fidelity" attribute.

## 3522 4.4.26 printer-up-time (integer(1:MAX))

3523 This MANDATORY Printer attribute indicates the amount of time (in seconds) that this instance of this  
3524 Printer implementation has been up and running. This value is used to populate the Job attributes "time-  
3525 at-creation", "time-at-processing", and "time-at-completed". These time values are all measured in  
3526 seconds and all have meaning only relative to this attribute, "printer-up-time". The value is a  
3527 monotonically increasing value starting from 1 when the Printer object is started-up (initialized, booted,  
3528 etc.).

3529 If the Printer object goes down at some value 'n', and comes back up, the implementation MAY:

- 3530 1. Know how long it has been down, and resume at some value greater than 'n', or
- 3531 2. Restart from 1.

3532  
3533 In the first case, the Printer SHOULD not tweak any existing related Job attributes ("time-at-creation",  
3534 "time-at-processing", and "time-at-completed"). In the second case, the Printer object SHOULD reset  
3535 those attributes to 0. If a client queries a time-related Job attribute and finds the value to be 0, the client  
3536 MUST assume that the Job was submitted in some life other than the Printer's current life.

## 3537 4.4.27 printer-current-time (dateTime)

3538 This Printer attribute indicates the current absolute wall-clock time. If an implementation supports this  
3539 attribute, then a client could calculate the absolute wall-clock time each Job's "time-at-creation", "time-at-  
3540 processing", and "time-at-completed" attributes by using both "printer-up-time" and this attribute,  
3541 "printer-current-time". If an implementation does not support this attribute, a client can only calculate  
3542 the relative time of certain events based on the MANDATORY "printer-up-time" attribute.

## 3543 4.4.28 multiple-operation-time-out (integer(1:MAX))

3544 This Printer attributes identifies how long (in seconds) the Printer object waits for additional Send-  
3545 Document or Send-URI operations to follow a still-open multi-document Job object before taking one of  
3546 the actions indicated in section 3.3.1.

## 3547 4.4.29 compression-supported (1setOf type3 keyword)

3548 This Printer attribute identifies the set of supported compression algorithms for document data.  
3549 Compression only applies to the document data; compression does not apply to the encoding of the IPP  
3550 operation itself. The supported values are used to validate the client supplied "compression" operation  
3551 attributes in Print-Job, Send-Document, and Send-URI requests.

3552 Standard values are :

3553 'none': no compression is used.  
3554 'deflate': ZIP public domain inflate/deflate) compression technology  
3555 'gzip' GNU zip compression technology described in RFC 1952 [RFC1952].  
3556 'compress': UNIX compression technology  
3557

#### 3558 4.4.30 job-k-octets-supported (rangeOfInteger(0:MAX))

3559 This Printer attribute specifies the upper and lower bounds of total sizes of jobs in K octets, i.e., in units  
3560 of 1024 octets. The supported values are used to validate the client supplied "job-k-octets" operation  
3561 attributes in create requests. The corresponding job description attribute "job-k-octets" is defined in  
3562 section 4.3.17.

#### 3563 4.4.31 job-impressions-supported (rangeOfInteger(0:MAX))

3564 This Printer attribute specifies the upper and lower bounds for the number of impressions per job. The  
3565 supported values are used to validate the client supplied "job-impressions" operation attributes in create  
3566 requests. The corresponding job description attribute "job-impressions" is defined in section 4.3.18.

#### 3567 4.4.32 job-media-sheets-supported (rangeOfInteger(0:MAX))

3568 This Printer attribute specifies the upper and lower bounds for the number of media sheets per job. The  
3569 supported values are used to validate the client supplied "job-media-sheets" operation attributes in create  
3570 requests. The corresponding Job attribute "job-media-sheets" is defined in section 4.3.19.

## 3571 5. Conformance

3572 This section describes conformance issues and requirements. This document introduces model entities  
3573 such as objects, operations, attributes, attribute syntaxes, and attribute values. These conformance  
3574 sections describe the conformance requirements which apply to these model entities.

### 3575 5.1 Client Conformance Requirements

3576 A conforming client SHALL support all MANDATORY operations as defined in this document. For  
3577 each attribute included in an operation request, a conforming client SHALL supply a value whose type  
3578 and value syntax conforms to the requirements of the Model document as specified in Sections 3 and 4.  
3579 A conforming client MAY supply any registered extensions and/or private extensions in an operation  
3580 request, as long as they meet the requirements in Section **Error! Reference source not found.**



3581 Otherwise, there are no conformance requirements placed on the user interfaces provided by IPP clients  
3582 or their applications. For example, one application might not allow an end user to submit multiple  
3583 documents per job, while another does. One application might first query a Printer object in order to  
3584 supply a graphical user interface (GUI) dialogue box with supported and default values whereas a  
3585 different implementation might not.

3586 When sending a request, an IPP client NEED NOT supply any attributes that are indicated as  
3587 OPTIONALLY supplied by the client.

3588 A client SHALL be able to accept any of the attribute syntaxes defined in Section 4.1, including their full  
3589 range, that may be returned to it in a response from a Printer object. For presentation purposes,  
3590 truncation of long attribute values is not recommended. A recommended approach would be for the  
3591 client implementation to allow the user to scroll through long attribute values.

3592 A query response may contain attribute groups, attributes, and values that the client does not expect.  
3593 Therefore, a client implementation MUST gracefully handle such responses and not refuse to inter-  
3594 operate with a conforming Printer that is returning extended registered or private attributes and/or  
3595 attribute values that conform to Section **Error! Reference source not found.** Clients may choose to  
3596 ignore any parameters, attributes, or values that ~~it does~~they do not understand.

## 3597 5.2 IPP Object Conformance Requirements

3598 This section specifies the conformance requirements for conforming implementations with respect to  
3599 objects, operations, and attributes.

### 3600 5.2.1 Objects

3601 Conforming implementations SHALL implement all of the model objects as defined in this specification in  
3602 the indicated sections:

3603 Section 2.1 - Printer Object

3604 Section 2.2 - Job Object

3605

### 3606 5.2.2 Operations

3607 Conforming IPP object implementations SHALL implement all of the MANDATORY model operations,  
3608 including mandatory responses, as defined in this specification in the indicated sections:

3609 For a Printer object:

3610 Print-Job (section 3.2.1)

MANDATORY

3611	Print-URI (section 3.2.2)	OPTIONAL
3612	Validate-Job (section 3.2.3)	MANDATORY
3613	Create-Job (section 3.2.4)	OPTIONAL
3614	Get-Printer-Attributes (section 3.2.5)	MANDATORY
3615	Get-Jobs (section 3.2.6)	MANDATORY
3616		
3617	For a Job object:	
3618	Send-Document (section 3.3.1)	OPTIONAL
3619	Send-URI (section 3.3.2)	OPTIONAL
3620	Cancel-Job (section 3.3.3)	MANDATORY
3621	Get-Job-Attributes (section 3.3.4)	MANDATORY
3622		

3623 Conforming IPP objects SHALL support all MANDATORY operation attributes and all values of such  
3624 attributes if so indicated in the description. Conforming IPP objects SHALL ignore all unsupported or  
3625 unknown operation attributes or operation attribute groups received in a request, but SHALL reject a  
3626 request that contains a supported operation attribute that contains an unsupported value.

3627 The following section on object attributes specifies the support required for object attributes.

### 3628 5.2.3 IPP Object Attributes

3629 Conforming IPP objects SHALL support all of the MANDATORY object attributes, as defined in this  
3630 specification in the indicated sections.

3631 If an object supports an attribute, it SHALL support only those values specified in this document or  
3632 through the extension mechanism described in section 5.2.4. It MAY support any non-empty subset of  
3633 these values. That is, it SHALL support at least one of the specified values and at most all of them.

### 3634 5.2.4 Extensions

3635 A conforming IPP object MAY support registered extensions and private extensions, as long as they meet  
3636 the requirements specified in Section **Error! Reference source not found.**

3637 For each attribute included in an operation response, a conforming IPP object SHALL return a value  
3638 whose type and value syntax conforms to the requirement of the Model document as specified in Sections  
3639 3 and 4.

3640 5.2.5 Attribute Syntaxes

3641 An IPP object SHALL be able to accept any of the attribute syntaxes defined in Section 4.1, including  
3642 their full range, in any operation in which a client may supply attributes or the system administrator may  
3643 configure attributes (by means outside the scope of IPP/1.0). Furthermore, an IPP object SHALL return  
3644 attributes to the client in operation responses that conform to the syntax specified in Section 4.1,  
3645 including their full range if supplied previously by a client.

3646 5.3 Charset and Natural Language Requirements

3647 All clients and IPP objects SHALL support the 'utf-8' charset as defined in section 4.1.7.

3648 IPP objects MUST be able to accept any client request which correctly uses the "attributes-natural-  
3649 language" operation attribute or the Natural Language Override mechanism on any individual attribute  
3650 whether or not the natural language is supported by the IPP object. If an IPP object supports a natural  
3651 language, then it MUST be able to translate (perhaps by table lookup) all generated 'text' or 'name'  
3652 attribute values into one of the supported languages (see section 3.1.4). That is, the IPP object that  
3653 supports a natural language NEED NOT be a general purpose translator of any arbitrary 'text' or 'name'  
3654 value supplied by the client into that natural language. However, the object MUST be able to translate  
3655 (automatically generate) any of its own attribute values and messages into that natural language.

3656 5.4 Security Conformance Requirements

3657 Conforming IPP Printer objects MAY support Transport Layer Security (TLS) access, support access  
3658 without TLS or support both means of access.

3659 Conforming IPP clients SHOULD support TLS access and non-TLS access. Note: This client  
3660 requirement to support both means that conforming IPP clients will be able to inter-operate with any IPP  
3661 Printer object.

3662 For a detailed discussion of security considerations and the IPP application security profile required for  
3663 TLS support, see section 8.

3664 6. IANA Considerations (registered and private extensions)

3665 This section describes how IPP can be extended to allow the following registered and private extensions  
3666 to IPP:

3667 1. keyword attribute values

3668 [2. enum attribute values](#)

3669 [3. attributes](#)

3670 [4. attribute syntaxes](#)

3671 [5. operations](#)

3672 [6. status codes](#)

3673

3674 [Registered and private extensions registered for use with IPP/1.0 are OPTIONAL for client and IPP](#)  
3675 [object conformance to the IPP/1.0 Model specification.](#)

3676 [These extension procedures are aligned with the guidelines as set forth by the IESG \[IANA-CON\].](#)  
3677 [Section 12 describes how to propose new registrations for consideration. IANA will reject registration](#)  
3678 [proposals that leave out required information or do not follow the appropriate format described in](#)  
3679 [Section 12.](#)

## 3680 6.1 [Typed 'keyword' and 'enum' Extensions](#)

3681 [IPP allows for 'keyword' and 'enum' extensions \(see sections 4.1.3 and 4.1.4\). This document uses](#)  
3682 [prefixes to the 'keyword' and 'enum' basic attribute syntax type in order to communicate extra information](#)  
3683 [to the reader through its name. This extra information need not be represented in the protocol because it](#)  
3684 [is unimportant to a client or Printer object. The list below describes the prefixes and their meaning.](#)

3685 ["type1": The IPP specification must be revised to add a new keyword or a new enum. No private](#)  
3686 [keywords or enums are allowed.](#)

3687  
3688 ["type2": Implementers can, at any time, add new keyword or enum values by proposing the complete](#)  
3689 [specification to IANA:](#)

3690  
3691 [iana@iana.org](#)

3692  
3693 [IANA will forward the registration proposal to the IPP Designated Expert who will review the](#)  
3694 [proposal with a mailing list that the Designated Expert keeps for this purpose. Initially, that list](#)  
3695 [will be the mailing list used by the IPP WG:](#)

3696  
3697 [ipp@pwg.org](#)

3698  
3699 [even after the IPP WG is disbanded as permitted by \[IANA-CON\]. The IPP Designated Expert is](#)  
3700 [appointed by the IESG Area Director responsible for IPP, according to \[IANA-CON\].](#)

3701

3702 When a type2 keyword or enum is approved, the IPP Designated Expert becomes the point of  
3703 contact for any future maintenance that might be required for that registration.

3704  
3705 "type3": Implementers can, at any time, add new keyword and enum values by submitting the  
3706 complete specification to IANA as for type2 who will forward the proposal to the IPP Designated  
3707 Expert. While no additional technical review is required, the IPP Designated Expert may, at  
3708 his/her discretion, forward the proposal to the same mailing list as for type2 registrations for  
3709 advice and comment.

3710  
3711 When a type3 keyword or enum is approved by the IPP Designated Expert, the original proposer  
3712 becomes the point of contact for any future maintenance that might be required for that  
3713 registration.

3714  
3715 For type2 and type3 keywords, the proposer includes the name of the keyword in the registration  
3716 proposal and the name is part of the technical review.

3717 After type2 and type3 enums specifications are approved, the IPP Designated Expert in consultation with  
3718 IANA assigns the next available enum number for each enum value.

3719 IANA will publish approved type2 and type3 keyword and enum attributes value registration  
3720 specifications in:

3721 <ftp://isi.edu/iana/assignments/ipp/attribute-values/xxx/yyy.txt>

3722 where xxx is the attribute name that specifies the initial values and yyy.txt is a descriptive file name that  
3723 contains one or more enums or keywords approved at the same time. For example, if several additional  
3724 enums for stapling are approved for use with the "finishings" attribute (and "finishings-default" and  
3725 "finishings-supported" attributes), IANA will publish the additional values in the file:

3726 <ftp://isi.edu/iana/assignments/ipp/attribute-values/finishings/stapling.txt>.

3727 Note: Some attributes are defined to be either 'type3 keywords' and 'name' which allows for attribute  
3728 values to be extended by a site administrator with administrator defined names. Such names are not  
3729 registered with IANA.

3730 By definition, each of the three types above assert some sort of registry or review process in order for  
3731 extensions to be considered valid. Each higher numbered level (1, 2, 3) tends to be decreasingly less  
3732 stringent than the previous level. Therefore, any typeN value MAY be registered using a process for  
3733 some typeM where M is less than N, however such registration is NOT REQUIRED. For example, a  
3734 type3 value MAY be registered in a type 1 manner (by being included in a future version of an IPP  
3735 specification), however, it is NOT REQUIRED.

3736 This specification defines keyword and enum values for all of the above types, including type3 keywords.

3737 For private (unregistered) keyword extensions, implementers SHOULD use keywords with a suitable  
3738 distinguishing prefix, such as "xxx-" where xxx is the (lowercase) fully qualified company name registered  
3739 with IANA for use in domain names [RFC1035]. For example, if the company XYZ Corp. had obtained  
3740 the domain name "XYZ.com", then a private keyword 'abc' would be: 'xyz.com-abc'.

3741 Note: RFC 1035 [RFC1035] indicates that while upper and lower case letters are allowed in domain  
3742 names, no significance is attached to the case. That is, two names with the same spelling but different  
3743 case are to be treated as if identical. Also, the labels in a domain name must follow the rules for  
3744 ARPANET host names: They must start with a letter, end with a letter or digit, and have as interior  
3745 characters only letters, digits, and hyphen. Labels must be 63 characters or less. Labels are separated by  
3746 the "." character.

3747 For private (unregistered) enum extension, implementers SHALL use values in the reserved integer range  
3748 which is 2\*\*30 to 2\*\*31-1.

## 3749 6.2 Attribute Extensibility

3750 Attribute names are type2 keywords. Therefore, new attributes may be registered and have the same  
3751 status as attributes in this document by following the type2 extension rules. For private (unregistered)  
3752 attribute extensions, implementers SHOULD use keywords with a suitable distinguishing prefix as  
3753 described in Section 6.1.

3754 IANA will publish approved attribute registration specifications as separate files:

3755 <ftp.isi.edu/iana/assignments/ipp/attributes/xxx-yyy.txt>

3756 where "xxx-yyy" is the new attribute name.

3757 If a new Printer object attribute is defined and its values can be affected by a specific document format, its  
3758 specification needs to contain the following sentence:

3759 "The value of this attribute returned in a Get-Printer-Attributes response MAY depend on the  
3760 "document-format" attribute supplied (see Section 3.2.5.1)."

3761 If the specification does not, then its value in the Get-Printer-Attributes response SHALL NOT depend  
3762 on the "document-format" supplied in the request. When a new Job Template attribute is registered, the  
3763 value of the Printer attributes MAY vary with "document-format" supplied in the request without the  
3764 specification having to indicate so.

3765 6.3 Attribute Syntax Extensibility

3766 Attribute syntaxes are like type2 enums. Therefore, new attribute syntaxes may be registered and have  
3767 the same status as attribute syntaxes in this document by following the type2 extension rules described in  
3768 Section 6.1. The value codes that identify each of the attribute syntaxes are assigned in the protocol  
3769 specification [IPP-PRO].

3770 For attribute syntaxes, the IPP Designated Expert in consultation with IANA assigns the next attribute  
3771 syntax code in the appropriate range as specified in [IPP-PRO]. IANA will publish approved attribute  
3772 syntax registration specifications as separate files:

3773 <ftp://isi.edu/iana/assignments/ipp/attribute-syntaxes/xxx-yyy.txt>

3774 where 'xxx-yyy' is the new attribute syntax name.

3775 6.4 Operation Extensibility

3776 Operations may also be registered following the type2 procedures described in Section 6.1, though major  
3777 new operations will usually be done by a new standards track RFC that augments this document. For  
3778 private (unregistered) operation extensions, implementers SHALL use the range for the "operation-id" in  
3779 requests specified in Section 4.4.13 "operations-supported" Printer attribute.

3780 For operations, the IPP Designated Expert in consultation with IANA assigns the next operation-id code  
3781 as specified in Section 4.4.13. IANA will publish approved operation registration specifications as  
3782 separate files:

3783 <ftp://isi.edu/iana/assignments/ipp/operations/Xxx-Yyy.txt>

3784 where "Xxx-Yyy" is the new operation name.

3785 6.5 Status Code Extensibility

3786 Operation status codes may also be registered following the type2 procedures described in Section 6.1.  
3787 The values for status codes are allocated in ranges as specified in Section 14 for each status code class:

3788 "informational" - Request received, continuing process

3789 "successful" - The action was successfully received, understood, and accepted

3790 "redirection" - Further action must be taken in order to complete the request

3791 "client-error" - The request contains bad syntax or cannot be fulfilled

3792 "server-error" - The IPP object failed to fulfill an apparently valid request

3793

3794 For private (unregistered) operation status code extensions, implementers SHALL use the top of each  
3795 range as specified in Section 14.

3796 For operation status codes, the IPP Designated Expert in consultation with IANA assigns the next status  
3797 code in the appropriate class range as specified in Section 14. IANA will publish approved status code  
3798 registration specifications as separate files:

3799 <ftp.isi.edu/iana/assignments/ipp/status-codes/xxx-yyy.txt>

3800 where "xxx-yyy" is the new operation status code keyword.

### 3801 6.6 Registration of MIME types/sub-types for document-formats

3802 The "document-format" attribute's syntax is 'mimeMediaType'. This means that valid values are Internet  
3803 Media Types (see Section 4.1.9). RFC 2045 [RFC2045] defines the syntax for valid Internet media  
3804 types. IANA is the registry for all Internet media types.

### 3805 6.7 Registration of charsets for use in 'charset' attribute values

3806 The "attributes-charset" attribute's syntax is 'charset'. This means that valid values are charsets names.  
3807 When a charset in the IANA registry has more than one name (alias), the name labeled as "(preferred  
3808 MIME name)", if present, SHALL be used (see Section 4.1.7). IANA is the registry for charsets  
3809 following the procedures of [IANA-CSa].

## 3810 7. Internationalization Considerations

3811 Some of the attributes have values that are text strings and names which are intended for human  
3812 understanding rather than machine understanding (see the 'text' and 'name' attribute syntaxes in Sections  
3813 4.1.1 and 4.1.2).

3814 In each operation request, the client

- 3815 - identifies the charset and natural language of the request which affects each supplied 'text' and 'name'
- 3816 attribute value, and
- 3817 - requests the charset and natural language for attributes returned by the IPP object in operation
- 3818 responses (as described in Section 3.1.4.1).

3819



3820 In addition, the client MAY separately and individually identify the Natural Language Override of a  
3821 supplied 'text' or 'name' attribute using the 'textWithLanguage' and 'nameWithLanguage' technique  
3822 described section 4.1.1.2 and 4.1.2.2 respectively.

3823 All IPP objects SHALL support the UTF-8 [RFC2044] charset in all 'text' and 'name' attributes  
3824 supported. If an IPP object supports more than the UTF-8 charset, the object SHALL convert between  
3825 them in order to return the requested charset to the client according to Section 3.1.4.2. If an IPP object  
3826 supports more than one natural language, the object SHOULD return 'text' and 'name' values in the  
3827 natural language requested where those values are generated by the Printer (see Section 3.1.4.1).

3828 For Printers that support multiple charsets and/or multiple natural languages in 'text' and 'name' attributes,  
3829 different jobs may have been submitted in differing charsets and/or natural languages. All responses  
3830 SHALL be returned in the charset requested by the client. However, the Get-Jobs operation uses the  
3831 'textWithLanguage' and 'nameWithLanguage' mechanism to identify the differing natural languages with  
3832 each job returned.

3833 The Printer object also has configured charset and natural language attributes. The client can query the  
3834 Printer object to determine the list of charsets and natural languages supported by the Printer object and  
3835 what the Printer object's configured values are. See the "charset-configured", "charset-supported",  
3836 "natural-language-configured", and "generated-natural-language-supported" Printer description attributes  
3837 for more details.

3838 The "charset-supported" attributed identifies the supported charsets. If a charset is supported, the IPP  
3839 object MUST be capable of converting to and from that charset into any other supported charset. In  
3840 many cases, an IPP object will support only one charset and it MUST be the UTF-8 charset.

3841 The "charset-configured" attribute identifies the one supported charset which is the native charset given  
3842 the current configuration of the IPP object (administrator defined).

3843 The "generated-natural-language-supported" attribute identifies the set of supported natural languages for  
3844 generated messages; it is not related to the set of natural languages that must be accepted for client  
3845 supplied 'text' and 'name' attributes. For client supplied 'text' and 'name' attributes, an IPP object MUST  
3846 accept ALL supplied natural languages. Just because a Printer object is currently configured to support  
3847 'en-us' natural language does not mean that the Printer object should reject a job if the client supplies a  
3848 job name that is in 'fr-ca'.

3849 The "natural-language-configured" attribute identifies the one supported natural language for generated  
3850 messages which is the native natural language given the current configuration of the IPP object  
3851 (administrator defined).

3852 Attributes of type 'text' and 'name' are populated from different sources. These attributes can be  
 3853 categorized into following groups (depending on the source of the attribute):

- 3854 1. Some attributes are supplied by the client (e.g., the client supplied "job-name", "document-name",  
 3855 and "requesting-user-name" operation attributes along with the corresponding Job object's "job-  
 3856 name" and "job-originating-user-name" attributes). The IPP object MUST accept these attributes  
 3857 in any natural language no matter what the set of supported languages for generated messages  
 3858 2. Some attributes are supplied by the system administrator (e.g., the Printer object's "printer-name"  
 3859 and "printer-location" attributes). These too can be in any natural language. If the natural  
 3860 language for these attributes is different than what a client requests, then they must be reported  
 3861 using the Natural Language Override mechanism.
- 3862 3. Some attributes are supplied by the device manufacturer (e.g., the Printer object's "printer-make-  
 3863 and-model" attribute). These too can be in any natural language. If the natural language for these  
 3864 attributes is different than what a client requests, then they must be reported using the Natural  
 3865 Language Override mechanism.
- 3866 4. Some attributes are supplied by the operator (e.g., the Job object's "job-message-from-operator"  
 3867 attribute). These too can be in any natural language. If the natural language for these attributes is  
 3868 different than what a client requests, then they must be reported using the Natural Language  
 3869 Override mechanism.
- 3870 5. Some attributes are generated by the IPP object (e.g., the Job object's "job-state-message"  
 3871 attribute, the Printer object's "printer-state-message" attribute, and the "status-message" operation  
 3872 attribute). These attributes can only be in one of the "generated-natural-language-supported"  
 3873 natural languages. If a client requests some natural language for these attributes other than one of  
 3874 the supported values, the IPP object SHOULD respond in using the value of the "natural-  
 3875 language-configured" attribute (using the Natural Language Override mechanism if needed).  
 3876

3877 The 'text' and 'name' attributes specified in this version of this document (additional ones will be  
 3878 registered according to the procedures in Section **Error! Reference source not found.**) are:

3879 Attributes	Source
3880 -----	-----
3881 Operation Attributes	
3882 job-name (name)	client
3883 document-name (name)	client
3884 requesting-user-name (name)	client
3885	
3886 Job Attributes:	
3887 job-name (name)	client or Printer object
3888 job-originating-user-name (name)	Printer object
3889 job-state-message (text)	Job or Printer object

3890	job-message-from-operator (text)	operator
3891		
3892	Printer Attributes:	
3893	printer-name (name)	administrator
3894	printer-location (text)	administrator
3895	printer-info (text)	administrator
3896	printer-make-and-model (text)	administrator or manufacturer
3897	printer-state-message (text)	Printer object
3898	printer-message-from-operator (text)	operator
3899		

## 3900 8. Security Considerations

3901 Some IPP objects MAY be deployed over protocol stacks that support Transport Layer Security (TLS)  
 3902 Version 1.0. Other IPP objects MAY be deployed over protocol stacks that do not support TLS. Some  
 3903 IPP objects MAY be deployed over both types of protocol stacks. Those IPP objects that support TLS,  
 3904 are capable of supporting mutual authentication as well as privacy of messages via multiple encryption  
 3905 schemes. TLS 1.0 also supports a backwards compatibility mode for negotiating down to SSL3 which  
 3906 leverages the vast installed base of SSL3 aware clients and servers. An important point about security  
 3907 related information for TLS access to an IPP object, is that the security-related parameters  
 3908 (authentication, encryption keys, etc.) are "out-of-band" to the actual IPP protocol.

3909 An IPP object that does not support TLS MAY elect to support a transport layer that provides other  
 3910 security mechanisms. For example, in a mapping of IPP over HTTP/1.1 [IPP-PRO], if the IPP object  
 3911 does not support TLS, HTTP still allows for client authentication.

3912 It is difficult to anticipate the security risks that might exist in any given IPP environment. For example, if  
 3913 IPP is used within a given corporation over a private network, the risks of exposing document data may  
 3914 be low enough that the corporation will choose not to use encryption on that data. However, if the  
 3915 connection between the client and the IPP object is over a public network, the client may wish to protect  
 3916 the content of the information during transmission through the network with encryption.

3917 Furthermore, the value of the information being printed may vary from one IPP environment to the next.  
 3918 Printing payroll checks, for example, would have a different value than printing public information from a  
 3919 file. There is also the possibility of denial-of-service attacks, but denial-of-service attacks against printing  
 3920 resources are not well understood and there is no published precedents regarding this scenario.

3921 Once the authenticated identity of the requester has been supplied to the IPP object, the object uses that  
 3922 identity to enforce any authorization policy that might be in place. For example, one site's policy might  
 3923 be that only the job owner is allowed to cancel a job. The details and mechanisms to set up a particular

3924 access control policy are not part of IPP/1.0, and must be established via some other type of  
3925 administrative or access control framework. However, there are operation status codes that allow an IPP  
3926 server to return information back to a client about any potential access control violations for an IPP  
3927 object.

3928 During a create operation, the client's identity is recorded in the Job object in an implementation-defined  
3929 attribute. This information can be used to verify a client's identity for subsequent operations on that Job  
3930 object in order to enforce any access control policy that might be in effect. See section 8.3 below for  
3931 more details.

3932 Since the security levels or the specific threats that any given IPP system administrator may be concerned  
3933 with cannot be anticipated, IPP MUST be capable of operating with different security mechanisms and  
3934 security policies as required by the individual installation. Security policies might vary from very strong,  
3935 to very weak, to none at all, and corresponding security mechanisms will be required. TLS Version 1.0  
3936 supports the type of negotiated levels of security required by most, if not all, potential IPP environments.  
3937 IPP environments that require no security can elect to deploy IPP objects that do not utilize the optional  
3938 TLS security mechanisms.

## 3939 8.1 Security Scenarios

3940 The following sections describe specific security attacks for IPP environments. Where examples are  
3941 provided they should be considered illustrative of the environment and not an exhaustive set. Not all of  
3942 these environments will necessarily be addressed in initial implementations of IPP.

### 3943 8.1.1 Client and Server in the Same Security Domain

3944 This environment is typical of internal networks where traditional office workers print the output of  
3945 personal productivity applications on shared work-group printers, or where batch applications print their  
3946 output on large production printers. Although the identity of the user may be trusted in this environment,  
3947 a user might want to protect the content of a document against such attacks as eavesdropping, replaying  
3948 or tampering.

### 3949 8.1.2 Client and Server in Different Security Domains

3950 Examples of this environment include printing a document created by the client on a publicly available  
3951 printer, such as at a commercial print shop; or printing a document remotely on a business associate's  
3952 printer. This latter operation is functionally equivalent to sending the document to the business associate  
3953 as a facsimile. Printing sensitive information on a Printer in a different security domain requires strong  
3954 security measures. In this environment authentication of the printer is required as well as protection  
3955 against unauthorized use of print resources. Since the document crosses security domains, protection

3956 against eavesdropping and document tampering are also required. It will also be important in this  
3957 environment to protect Printers against "spamming" and malicious document content.

### 3958 8.1.3 Print by Reference

3959 When the document is not stored on the client, printing can be done by reference. That is, the print  
3960 request can contain a reference, or pointer, to the document instead of the actual document itself.  
3961 Standard methods currently do not exist for remote entities to "assume" the credentials of a client for  
3962 forwarding requests to a 3rd party. It is anticipated that Print-By-Reference will be used to access  
3963 "public" documents and that sophisticated methods for authenticating "proxies" will not be specified for  
3964 version 1 of IPP.

### 3965 8.2 URIs for TLS and non-TLS Access

3966 As described earlier, an IPP object can support TLS access, non-TLS access, or both. The "printer-uri-  
3967 supported" attribute contains the Printer object's URI(s). Its companion attribute, "uri-security-  
3968 supported", identifies the security mechanism used for each URI listed in the "printer-uri-supported"  
3969 attribute. For each Printer operation request, a client SHALL supply only one URI in the "printer-uri"  
3970 operation attribute. In other words, even though the Printer supports more than one URI, the client only  
3971 interacts with the Printer object using one of its URIs. This duality is not needed for Job objects, since the  
3972 Printer objects is the factory for Job objects, and the Printer object will generate the correct URI for new  
3973 Job objects depending on the Printer object's security configuration.

### 3974 8.3 The "requesting-user-name" (name(MAX)) Operation Attribute

3975 Each operation SHALL specify the user who is performing the operation in both of the following two  
3976 ways:

- 3977 1) via the MANDATORY "requesting-user-name" operation attribute that a client SHOULD supply  
3978 in all operations. The client SHALL obtain the value for this attribute from an environmental or  
3979 network login name for the user, rather than allowing the user to supply any value. If the client  
3980 does not supply a value for "requesting-user-name", the printer SHALL assume that the client is  
3981 supplying some anonymous name, such as "anonymous".
- 3982 2) via an authentication mechanism of the underlying transport which may be configured to give no  
3983 authentication information.

3984  
3985 There are six cases to consider:

- 3986 a) the authentication mechanism gives no information, and the client doesn't specify "requesting-  
3987 user-name".
- 3988 b) the authentication mechanism gives no information, but the client specifies "requesting-user-  
3989 name".
- 3990 c) the authentication mechanism specifies a user which has no human readable representation, and the  
3991 client doesn't specify "requesting-user-name".
- 3992 d) the authentication mechanism specifies a user which has no human readable representation, but the  
3993 client specifies "requesting-user-name".
- 3994 e) the authentication mechanism specifies a user which has a human readable representation. The  
3995 Printer object ignores the "requesting-user-name".
- 3996 f) the authentication mechanism specifies a user who is trusted and whose name means that the value  
3997 of the "requesting-user-name", which MUST be present, is treated as the authenticated name.  
3998

3999 Note: Case "f" is intended for a tightly coupled gateway and server to work together so that the "user"  
4000 name is able to be that of the gateway client and not that of the gateway. Because most, if not all, system  
4001 vendors will initially implement IPP via a gateway into their existing print system, this mechanism is  
4002 necessary unless the authentication mechanism allows a gateway (client) to act on behalf of some other  
4003 client.

4004 The user-name has two forms:

- 4005 - one that is human readable: it is held in the MANDATORY "job-originating-user-name" Job  
4006 Description attribute which is set during the job creation operations. It is used for presentation  
4007 only, such as returning in queries or printing on start sheets
- 4008 - one for authorization: it is held in an undefined (by IPP) Job object attribute which is set by the job  
4009 creation operation. It is used to authorize other operations, such as Send-Document, Send-URI,  
4010 Cancel-Job, to determine the user when the my-jobs' attribute is specified with Get-Jobs, and to  
4011 limit what attributes and values to return with Get-Job-Attributes and Get-Jobs.  
4012

4013 The human readable user name:

- 4014 - is the value of the "requesting-user-name" for cases b, d and f.  
4015 - comes from the authentication mechanism for case e  
4016 - is some anonymous name, such as "anonymous" for cases a and c.  
4017

4018 The user name used for authorization:

- 4019 - is the value of the "requesting-user-name" for cases b and f.  
4020 - comes from the authentication mechanism for cases c, d and e  
4021 - is some anonymous name, such as "anonymous" for case a.  
4022

4023 The essence of these rules for resolving conflicting sources of user-names is that a printer implementation  
4024 is free to pick either source as long as it achieves consistent results. That is, if a user uses the same path  
4025 for a series of requests, the requests **MUST** appear to come from the same user from the standpoint of  
4026 both the human-readable user name and the user name for authorization. This rule **MUST** continue to  
4027 apply even if a request could be authenticated by two or more mechanisms. It doesn't matter which of  
4028 several authentication mechanisms a Printer uses as long as it achieves consistent results. If a client uses  
4029 more than one authentication mechanism, it is recommended that an administrator make all credentials  
4030 resolve to the same user and user-name as much as possible.

#### 4031 8.4 Restricted Queries

4032 In many IPP operations, a client supplies a list of attributes to be returned in the response. For security  
4033 reasons, an IPP object may be configured not to return all attributes (or all values) that a client requests.  
4034 The job attributes returned **MAY** depend on whether the requesting user is the same as the user that  
4035 submitted the job. The IPP object **MAY** even return none of the requested attributes. In such cases, the  
4036 status returned is the same as if the object had returned all requested attributes. The client cannot tell by  
4037 such a response whether the requested attribute was present or absent on the object.

#### 4038 8.5 IPP Security Application Profile for TLS

4039 The IPP application profile for TLS follows the standard "Mandatory Cipher Suites" requirement as  
4040 documented in the TLS specification [TLS]. Client implementations **MUST NOT** assume any other  
4041 cipher suites are supported by an IPP Printer object.

4042 If a conforming IPP object supports TLS, it **MUST** implement and support the "Mandatory Cipher  
4043 Suites" as specified in the TLS specification and **MAY** support additional cipher suites.

4044 A conforming IPP client **SHOULD** support TLS including the "Mandatory Cipher Suites" as specified in  
4045 the TLS specification. A conforming IPP client **MAY** support additional cipher suites.

4046 It is possible that due to certain government export restrictions some non-compliant versions of this  
4047 extension could be deployed. Implementations wishing to inter-operate with such non-compliant versions  
4048 **MAY** offer the TLS\_DHE\_DSS\_EXPORT\_WITH\_DES40\_CBC\_SHA mechanism. However, since 40  
4049 bit ciphers are known to be vulnerable to attack by current technology, any client which activates a 40 bit  
4050 cipher **MUST NOT** indicate to the user that the connection is completely secure from eavesdropping.

## 4051 9. References

## 4052 [ASCII]

4053 Coded Character Set - 7-bit American Standard Code for Information Interchange (ASCII),  
4054 ANSI X3.4-1986. This standard is the specification of the US-ASCII charset.

## 4055 [CS-POL]

4056 H. Alvestrand, "IETF Policy on Character Sets and Languages, work in progress <draft-  
4057 alvestrand-charset-policy-01.txt>, August 29, 1997.

## 4058 [HTPP]

4059 J. Barnett, K. Carter, R. DeBry, "Initial Draft - Hypertext Printing Protocol - HTPP/1.0",  
4060 October 1996, <ftp://ftp.pwg.org/pub/pwg/ipp/historic/http/-overview.ps.gz>

## 4061 [IANA-CON]

4062 Narte, T. and Alverstran, H.T.: Guidelines for Writing an IANA Considerations Section in RFCs,  
4063 Work in Progress, <draft-ietf-iana-considerations-014.txt>, [May 21, 1998](#).

## 4064 [IANA-CS]

4065 IANA Registry of Coded Character Sets: <ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets>

## 4066 [IANA-CSa]

4067 N. Freed, J. Postel: IANA CharSet Registration Procedures, Work in Progress (draft-freed-  
4068 charset-reg-02.txt).

## 4069 [IANA-MT]

4070 IANA Registry of Media Types: <ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/>

## 4071 [IPP-PRO]

4072 Herriot, R., Butler, S., Moore, P., Tuner, R., " Internet Printing Protocol/1.0: Protocol  
4073 Specifications", <draft-ietf-ipp-pro-05.txt>, November, 1997.

## 4074 [IPP-RAT]

4075 Zilles, S., "Rationale for the Structure and Model and Protocol for the Internet Printing Protocol",  
4076 <draft-ietf-ipp-rat-02.txt>, November, 1997.

## 4077 [IPP-REQ]

4078 Wright, D., "Requirements for an Internet Printing Protocol", <draft-ietf-ipp-req-.txt>, November,  
4079 1997.



- 4080 [ISO10646-1]  
4081 ISO/IEC 10646-1:1993, "Information technology -- Universal Multiple-Octet Coded Character  
4082 Set (UCS) - Part 1: Architecture and Basic Multilingual Plane, JTC1/SC2."
- 4083 [ISO8859-1]  
4084 ISO/IEC 8859-1:1987, "Information technology -- 8-bit One-Byte Coded Character Set - Part 1:  
4085 Latin Alphabet Nr 1", 1987, JTC1/SC2.
- 4086 [ISO10175]  
4087 ISO/IEC 10175 Document Printing Application (DPA), June 1996.
- 4088 [LDPA]  
4089 T. Hastings, S. Isaacson, M. MacKay, C. Manros, D. Taylor, P. Zehler, "LDPA - Lightweight  
4090 Document Printing Application", October 1996,  
4091 <ftp://ftp.pwg.org/pub/pwg/ipp/historic/ldpa/ldpa8.pdf.gz>
- 4092 [P1387.4]  
4093 Kirk, M. (editor), POSIX System Administration - Part 4: Printing Interfaces, POSIX 1387.4 D8,  
4094 1994.
- 4095 [PSIS] Herriot, R. (editor), X/Open A Printing System Interoperability Specification (PSIS), August  
4096 1995.
- 4097 [PWG]  
4098 Printer Working Group, <http://www.pwg.org>.
- 4099 [RFC1035]  
4100 P. Mockapetris, "DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION", RFC  
4101 1035, November 1987.
- 4102 [RFC1179]  
4103 McLaughlin, L. III, (editor), "Line Printer Daemon Protocol" RFC 1179, August 1990.
- 4104 [RFC1630]  
4105 T. Berners-Lee, "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression  
4106 of Names and Addresses of Objects on the Network as used in the World-Wide Web", RFC 1630,  
4107 June 1994.
- 4108 [RFC1738]  
4109 Berners-Lee, T., Masinter, L., McCahill, M. , "Uniform Resource Locators (URL)", RFC 1738,  
4110 December, 1994.

- 4111 [RFC1759]  
4112 Smith, R., Wright, F., Hastings, T., Zilles, S., and Gyllenskog, J., "Printer MIB", RFC 1759,  
4113 March 1995.
- 4114 [RFC1766]  
4115 H. Alvestrand, " Tags for the Identification of Languages", RFC 1766, March 1995.
- 4116 [RFC1903]  
4117 J. Case, et.al., "Textual Conventions for Version 2 of the Simple Network Management Protocol  
4118 (SNMP v2)" RFC 1903, January 1996.
- 4119 [RFC1952]  
4120 P. Deutsch, "GZIP file format specification version 4.3", RFC 1952, May 1996.
- 4121 [RFC2044]  
4122 F. Yergeau, "UTF-8, a transformation format of Unicode and ISO 10646", RFC 2044, October  
4123 1996.
- 4124 [RFC2068]  
4125 R. Fielding, J. Gettys, J. Mogul, H. Frystyk, T. Berners-Lee, "Hypertext Transfer Protocol -  
4126 HTTP/1.1", RFC 2068, January 1997
- 4127 [RFC2069]  
4128 J. Franks, P. Hallam-Baker, J. Hostetler, P. Leach, A. Luotonen, E. Sink, L. Stewart, "An  
4129 Extension to HTTP: Digest Access Authentication", RFC-2069, Jan 1997.
- 4130 [RFC2045]  
4131 N. Fried, N. Borenstein, ", Multipurpose Internet Mail Extensions (MIME) Part One: Format of  
4132 Internet Message Bodies " RFC 2045, November 1996.
- 4133 [RFC2046]  
4134 Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types. N. Freed & N.  
4135 Borenstein. November 1996. (Obsoletes RFC1521, RFC1522, RFC1590), RFC 2046.
- 4136 [RFC2048]  
4137 Multipurpose Internet Mail Extension (MIME) Part Four: Registration Procedures. N. Freed, J.  
4138 Klensin & J. Postel. November 1996. (Format: TXT=45033 bytes) (Obsoletes RFC1521,  
4139 RFC1522, RFC1590) (Also BCP0013), RFC 2048.
- 4140 [RFC2119]  
4141 S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels", RFC 2119 , March  
4142 1997

4143 [SWP]  
4144 P. Moore, B. Jahromi, S. Butler, "Simple Web Printing SWP/1.0", May 7, 1997,  
4145 [ftp://ftp.pwg.org/pub/pwg/ipp/new\\_PRO/swp9705.pdf](ftp://ftp.pwg.org/pub/pwg/ipp/new_PRO/swp9705.pdf)

## 4146 10. Copyright Notice

4147 This document and translations of it may be copied and furnished to others, and derivative works that  
4148 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published  
4149 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright  
4150 notice and this paragraph are included on all such copies and derivative works. However, this document  
4151 itself may not be modified in any way, such as by removing the copyright notice or references to the  
4152 Internet Society or other Internet organizations, except as needed for the purpose of developing Internet  
4153 standards in which case the procedures for copyrights defined in the Internet Standards process must be  
4154 followed, or as required to translate it into languages other than English.

4155 The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its  
4156 successors or assigns.

4157 This document and the information contained herein is provided on an "AS IS" basis and THE  
4158 INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL  
4159 WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY  
4160 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY  
4161 RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A  
4162 PARTICULAR PURPOSE.

## 4163 11. Author's Address

4164 Scott A. Isaacson (Editor)  
4165 Novell, Inc.  
4166 122 E 1700 S  
4167 Provo, UT 84606

4168  
4169 Phone: 801-861-7366  
4170 Fax: 801-861-2517  
4171 e-mail: [sisaacson@novell.com](mailto:sisaacson@novell.com)

4172  
4173 Tom Hastings  
4174 Xerox Corporation

4175 701 S. Aviation Blvd.  
4176 El Segundo, CA 90245  
4177  
4178 Phone: 310-333-6413  
4179 Fax: 310-333-5514  
4180 e-mail: [hastings@cp10.es.xerox.com](mailto:hastings@cp10.es.xerox.com)  
4181  
4182 Robert Herriot  
4183 Sun Microsystems Inc.  
4184 901 San Antonio.Road, MPK-17  
4185 Palo Alto, CA 94303  
4186  
4187 Phone: 650-786-8995  
4188 Fax: 650-786-7077  
4189 e-mail: [robert.herriot@eng.sun.com](mailto:robert.herriot@eng.sun.com)  
4190  
4191 Roger deBry  
4192 HUC/003G  
4193 IBM Corporation  
4194 P.O. Box 1900  
4195 Boulder, CO 80301-9191  
4196  
4197 Phone: (303) 924-4080  
4198 Fax: (303) 924-9889  
4199 e-mail: [debry@vnet.ibm.com](mailto:debry@vnet.ibm.com)  
4200  
4201 Patrick Powell  
4202 San Diego State University  
4203 9475 Chesapeake Dr., Suite D  
4204 San Diego, CA 95123  
4205  
4206 Phone: (619) 874-6543  
4207 Fax: (619) 279-8424  
4208 e-mail: [papowell@sdsu.edu](mailto:papowell@sdsu.edu)  
4209  
4210 IPP Mailing List: [ipp@pwg.org](mailto:ipp@pwg.org)  
4211 IPP Mailing List Subscription: [ipp-request@pwg.org](mailto:ipp-request@pwg.org)  
4212 IPP Web Page: <http://www.pwg.org/ipp/>  
4213

4214 Implementers of this specification are encouraged to join IPP Mailing List in order to participate in any  
4215 discussions of clarification issues and review of registration proposals for additional attributes and values.  
4216

4217 Other Participants:

4218 Chuck Adams - Tektronix  
4219 Jeff Barnett - IBM  
4220 Ron Bergman - Dataproducts Corp.  
4221 Sylvan Butler, HP  
4222 Keith Carter, IBM Corporation  
4223 Jeff Copeland - QMS  
4224 Andy Davidson - Tektronix  
4225 Mabry Dozier - QMS  
4226 Lee Farrell - Canon Information Systems  
4227 Steve Gebert - IBM  
4228 Babek Jahromi, Microsoft  
4229 David Kellerman - Northlake Software  
4230 Rick Landau - Digital  
4231 Greg LeClair - Epson  
4232 Harry Lewis - IBM  
4233 Pete Loya - HP  
4234 Ray Lutz - Cognisys  
4235 Mike MacKay, Novell, Inc.  
4236 Carl-Uno Manros, Xerox, Corp.  
4237 Jay Martin - Underscore  
4238 Stan McConnell - Xerox  
4239 Ira McDonald, High North Inc.  
4240 Paul Moore, Microsoft  
4241 Tetsuya Morita - Ricoh  
4242 Yuichi Niwa - Ricoh  
4243 Pat Nogay - IBM  
4244 Ron Norton - Printronics  
4245 Bob Pentecost - HP  
4246 Rob Rhoads - Intel  
4247 Xavier Riley - Xerox, Corp.  
4248 David Roach - Unisys  
4249 Stuart Rowley, Kyocera  
4250 Hiroyuki Sato - Canon  
4251 Bob Setterbo - Adobe  
4252 Devon Taylor, Novell, Inc.  
4253 Mike Timperman - Lexmark

4254 Randy Turner - Sharp  
4255 Atsushi Yuki - Kyocera  
4256 Rick Yardumian - Xerox, Corp.  
4257 Lloyd Young - Lexmark  
4258 Bill Wagner - DPI  
4259 Jim Walker - DAZEL  
4260 Chris Wellens - Interworking Labs  
4261 Rob Whittle - Novell  
4262 Don Wright - Lexmark  
4263 Peter Zehler, Xerox, Corp.  
4264 Steve Zilles, Adobe

4265

4266 12. Formats for IPP Registration Proposals

4267 This section specifies the required information and the formats for proposing registrations of extensions  
4268 to IPP as provided in Section 6 for:

4269

4270 1. type2 'keyword' attribute values4271 2. type3 'keyword' attribute values4272 3. type2 'enum' attribute values4273 4. type3 'enum' attribute values4274 5. attributes4275 6. attribute syntaxes4276 7. operations4277 8. status codes4278 12.1 Type2 keyword attribute values registration4279 Type of registration: type2 keyword attribute value4280 Name of attribute to which this keyword specification is to be added:4281 Proposed keyword name of this keyword value:4282 Specification of this keyword value (follow the style of IPP Model Section 4.1.3):4283 Name of proposer:4284 Address of proposer:4285 Email address of proposer:

4286

4287 Note: For type2 keywords, the Designated Expert will be the point of contact for the approved  
4288 registration specification, if any maintenance of the registration specification is needed.

4289 12.2 Type3 keyword attribute values registration4290 Type of registration: type3 keyword attribute value4291 Name of attribute to which this keyword specification is to be added:4292 Proposed keyword name of this keyword value:4293 Specification of this keyword value (follow the style of IPP Model Section 4.1.3):4294 Name of proposer:4295 Address of proposer:4296 Email address of proposer:

4297

4298 Note: For type3 keywords, the proposer will be the point of contact for the approved registration  
4299 specification, if any maintenance of the registration specification is needed.

### 4300 12.3 Type2 enum attribute values registration

4301 Type of registration: type2 enum attribute value

4302 Name of attribute to which this enum specification is to be added:

4303 Keyword symbolic name of this enum value:

4304 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4305 Specification of this enum value (follow the style of IPP Model Section 4.1.4):

4306 Name of proposer:

4307 Address of proposer:

4308 Email address of proposer:

4309

4310 Note: For type2 enums, the Designated Expert will be the point of contact for the approved registration  
4311 specification, if any maintenance of the registration specification is needed.

### 4312 12.4 Type3 enum attribute values registration

4313 Type of registration: type3 enum attribute value

4314 Name of attribute to which this enum specification is to be added:

4315 Keyword symbolic name of this enum value:

4316 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4317 Specification of this enum value (follow the style of IPP Model Section 4.1.4):

4318 Name of proposer:

4319 Address of proposer:

4320 Email address of proposer:

4321

4322 Note: For type3 enums, the proposer will be the point of contact for the approved registration  
4323 specification, if any maintenance of the registration specification is needed.

### 4324 12.5 Attribute registration

4325 Type of registration: attribute

4326 Proposed keyword name of this attribute:

4327 Types of attribute (Operation, Job Template, Job Description, Printer Description):

4328 Operations to be used with if the attribute is an operation attribute:

4329 Object (Job, Printer, etc. if bound to an object):

4330 Attribute syntax(es) (include 1setOf and range as in Section 4.2):

4331 If attribute syntax is 'keyword' or 'enum', is it type2 or type3:

4332 Specification of this attribute (follow the style of IPP Model Section 4.2):



4333 Name of proposer:

4334 Address of proposer:

4335 Email address of proposer:

4336

4337 Note: For attributes, the IPP Designated Expert will be the point of contact for the approved registration  
4338 specification, if any maintenance of the registration specification is needed.

#### 4339 12.6 Attribute Syntax registration

4340 Type of registration: attribute syntax

4341 Proposed name of this attribute syntax:

4342 Type of attribute syntax (integer, octetString, character-string, see [IPP-PRO]):

4343 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4344 Specification of this attribute (follow the style of IPP Model Section 4.1):

4345 Name of proposer:

4346 Address of proposer:

4347 Email address of proposer:

4348

4349 Note: For attribute syntaxes, the IPP Designated Expert will be the point of contact for the approved  
4350 registration specification, if any maintenance of the registration specification is needed.

#### 4351 12.7 Operation registration

4352 Type of registration: operation

4353 Proposed name of this operation:

4354 Numeric operation-id value (to be assigned by the IPP Designated Expert in consultation with IANA):

4355 Object Target (Job, Printer, etc. that operation is upon):

4356 Specification of this attribute (follow the style of IPP Model Section 3):

4357 Name of proposer:

4358 Address of proposer:

4359 Email address of proposer:

4360

4361 Note: For operations, the IPP Designated Expert will be the point of contact for the approved  
4362 registration specification, if any maintenance of the registration specification is needed.

#### 4363 12.8 Status code registration

4364 Type of registration: status code

4365 Keyword symbolic name of this status code value:

4366 Numeric value (to be assigned by the IPP Designated Expert in consultation with IANA):

4367 Operations that this status code may be used with:

4368 Specification of this status code (follow the style of IPP Model Section 14 APPENDIX B: Status Codes  
4369 and Suggested Status Code Messages):

4370 Name of proposer:

4371 Address of proposer:

4372 Email address of proposer:

4373

4374 Note: For status codes, the Designated Expert will be the point of contact for the approved registration  
4375 specification, if any maintenance of the registration specification is needed.

### 4376 13. APPENDIX A: Terminology

4377 This specification uses the terminology defined in this section.

#### 4378 ~~12.1~~13.1 Conformance Terminology

4379 The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",  
4380 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be  
4381 interpreted as described in RFC 2119 [RFC2119]. The sections below reiterate these definitions and  
4382 include some additional ones.

##### 4383 ~~12.1.1~~13.1.1 MUST

4384 This word, or the terms "REQUIRED", "SHALL" or "MANDATORY", means that the definition is an  
4385 absolute requirement of the specification.

##### 4386 ~~12.1.2~~13.1.2 MUST NOT

4387 This phrase, or the phrase "SHALL NOT", means that the definition is an absolute prohibition of the  
4388 specification.

##### 4389 ~~12.1.3~~13.1.3 SHOULD

4390 This word, or the adjective "RECOMMENDED", means that there may exist valid reasons in particular  
4391 circumstances to ignore a particular item, but the full implications must be understood and carefully  
4392 weighed before choosing a different course.

4393 ~~12.1.4~~13.1.4 SHOULD NOT

4394 This phrase, or the phrase "NOT RECOMMENDED" means that there may exist valid reasons in  
4395 particular circumstances when the particular behavior is acceptable or even useful, but the full  
4396 implications should be understood and the case carefully weighed before implementing any behavior  
4397 described with this label.

4398 ~~12.1.5~~13.1.5 MAY

4399 This word, or the adjective "OPTIONAL", means that an item is truly optional. One vendor may choose  
4400 to include the item because a particular marketplace requires it or because the vendor feels that it  
4401 enhances the product while another vendor may omit the same item. An implementation which does not  
4402 include a particular option MUST be prepared to inter-operate with another implementation which does  
4403 include the option, though perhaps with reduced functionality. In the same vein an implementation which  
4404 does include a particular option MUST be prepared to inter-operate with another implementation which  
4405 does not include the option (except, of course, for the feature the option provides.)

4406 ~~12.1.6~~13.1.6 NEED NOT

4407 The verb "NEED NOT" indicates an action that the subject of the sentence does not have to implement in  
4408 order to claim conformance to the standard. The verb "NEED NOT" is used instead of "MAY NOT"  
4409 since "MAY NOT" sounds like a prohibition.

4410 ~~12.2~~13.2 Model Terminology4411 ~~12.2.1~~13.2.1 Keyword

4412 Keywords are used within this document as identifiers of semantic entities within the abstract model (see  
4413 section 4.1.3). Attribute names, some attribute values, attribute syntaxes, and attribute group names are  
4414 represented as keywords.

4415 ~~12.2.2~~13.2.2 Attributes

4416 An attribute is an item of information that is associated with an instance of an IPP object. An attribute  
4417 consists of an attribute name and one or more attribute values. Each attribute has a specific attribute  
4418 syntax. All object attributes are defined in section 4 and all operation attributes are defined in section 3.

4419 Job Template Attributes are described in section 4.2. The client optionally supplies Job Template  
4420 attributes in a create request (operation requests that create Job objects). The Printer object has  
4421 associated attributes which define supported and default values for the Printer.

4422 ~~12.2.2.1~~13.2.2.1 Attribute Name

4423 Each attribute is uniquely identified in this document by its attribute name. An attribute name is a  
4424 keyword. The keyword attribute name is given in the section header describing that attribute. In running  
4425 text in this document, attribute names are indicated inside double quotation marks (") where the  
4426 quotation marks are not part of the keyword itself.

4427 ~~12.2.2.2~~13.2.2.2 Attribute Group Name

4428 Related attributes are grouped into named groups. The name of the group is a keyword. The group  
4429 name may be used in place of naming all the attributes in the group explicitly. Attribute groups are  
4430 defined in section 3.

4431 ~~12.2.2.3~~13.2.2.3 Attribute Value

4432 Each attribute has one or more values. Attribute values are represented in the syntax type specified for  
4433 that attribute. In running text in this document, attribute values are indicated inside single quotation  
4434 marks ('), whether their attribute syntax is keyword, integer, text, etc. where the quotation marks are not  
4435 part of the value itself.

4436 ~~12.2.2.4~~13.2.2.4 Attribute Syntax

4437 Each attribute is defined using an explicit syntax type. In this document, each syntax type is defined as a  
4438 keyword with specific meaning. The protocol specification document [IPP-PRO] indicates the actual  
4439 "on-the-wire" encoding rules for each syntax type. Attribute syntax types are defined in section 4.1.

4440 ~~12.2.3~~13.2.3 Supports

4441 By definition, a Printer object supports an attribute only if that Printer object responds with the  
4442 corresponding attribute populated with some value(s) in a response to a query for that attribute. A  
4443 Printer object supports an attribute value if the value is one of the Printer object's "supported values"  
4444 attributes. The device behind a Printer object may exhibit a behavior that corresponds to some IPP  
4445 attribute, but if the Printer object, when queried for that attribute, doesn't respond with the attribute, then  
4446 as far as IPP is concerned, that implementation does not support that feature. If the Printer object's "xxx-  
4447 supported" attribute is not populated with a particular value (even if that value is a legal value for that  
4448 attribute), then that Printer object does not support that particular value.

4449 A conforming implementation SHALL support all MANDATORY attributes. However, even for  
4450 MANDATORY attributes, conformance to IPP does not mandate that all implementations support all  
4451 possible values representing all possible job processing behaviors and features. For example, if a given

4452 instance of a Printer supports only certain document formats, then that Printer responds with the  
4453 "document-format-supported" attribute populated with a set of values, possibly only one, taken from the  
4454 entire set of possible values defined for that attribute. This limited set of values represents the Printer's set  
4455 of supported document formats. Supporting an attribute and some set of values for that attribute enables  
4456 IPP end users to be aware of and make use of those features associated with that attribute and those  
4457 values. If an implementation chooses to not support an attribute or some specific value, then IPP end  
4458 users would have no ability to make use of that feature within the context of IPP itself. However, due to  
4459 existing practice and legacy systems which are not IPP aware, there might be some other mechanism  
4460 outside the scope of IPP to control or request the "unsupported" feature (such as embedded instructions  
4461 within the document data itself).

4462 For example, consider the "finishings-supported" attribute.

- 4463 1) If a Printer object is not physically capable of stapling, the "finishings-supported" attribute MUST  
4464 NOT be populated with the value of 'staple'.
- 4465 2) A Printer object is physically capable of stapling, however an implementation chooses not to  
4466 support stapling in the IPP "finishings" attribute. In this case, 'staple' SHALL NOT be a value in  
4467 the "finishings-supported" Printer object attribute. Without support for the value 'staple', an IPP  
4468 end user would have no means within the protocol itself to request that a Job be stapled.  
4469 However, an existing document data formatter might be able to request that the document be  
4470 stapled directly with an embedded instruction within the document data. In this case, the IPP  
4471 implementation does not "support" stapling, however the end user is still able to have some  
4472 control over the stapling of the completed job.
- 4473 3) A Printer object is physically capable of stapling, and an implementation chooses to support  
4474 stapling in the IPP "finishings" attribute. In this case, 'staple' SHALL be a value in the "finishings-  
4475 supported" Printer object attribute. Doing so, would enable end users to be aware of and make  
4476 use of the stapling feature using IPP attributes.

4477  
4478 Even though support for Job Template attributes by a Printer object is OPTIONAL, it is  
4479 RECOMMENDED that if the device behind a Printer object is capable of realizing any feature or  
4480 function that corresponds to an IPP attribute and some associated value, then that implementation  
4481 SHOULD support that IPP attribute and value.

4482 The set of values in any of the supported value attributes is set (populated) by some administrative  
4483 process or automatic sensing mechanism that is outside the scope of IPP. For administrative policy and  
4484 control reasons, an administrator may choose to make only a subset of possible values visible to the end  
4485 user. In this case, the real output device behind the IPP Printer abstraction may be capable of a certain  
4486 feature, however an administrator is specifying that access to that feature not be exposed to the end user  
4487 through the IPP protocol. Also, since a Printer object may represent a logical print device (not just a  
4488 physical device) the actual process for supporting a value is undefined and left up to the implementation.

4489 However, if a Printer object supports a value, some manual human action may be needed to realize the  
4490 semantic action associated with the value, but no end user action is required.

4491 For example, if one of the values in the "finishings-supported" attribute is 'staple', the actual process  
4492 might be an automatic staple action by a physical device controlled by some command sent to the device.  
4493 Or, the actual process of stapling might be a manual action by an operator at an operator attended Printer  
4494 object.

4495 For another example of how supported attributes function, consider a system administrator who desires  
4496 to control all print jobs so that no job sheets are printed in order to conserve paper. To force no job  
4497 sheets, the system administrator sets the only supported value for the "job-sheets-supported" attribute to  
4498 'none'. In this case, if a client requests anything except 'none', the create request is rejected or the "job-  
4499 sheets" value is ignored (depending on the value of "ipp-attribute-fidelity"). To force the use of job  
4500 start/end sheets on all jobs, the administrator does not include the value 'none' in the "job-sheets-  
4501 supported" attribute. In this case, if a client requests 'none', the create request is rejected or the "job-  
4502 sheets" value is ignored (again depending on the value of "ipp-attribute-fidelity").

4503 ~~12.2.4~~13.2.4 print-stream page

4504 A "print-stream page" is a page according to the definition of pages in the language used to express the  
4505 document data.

4506 ~~12.2.5~~13.2.5 impression

4507 An "impression" is the image (possibly many print-stream pages in different configurations) imposed onto  
4508 a single media page.

4509 ~~13.14.~~ APPENDIX B: Status Codes and Suggested Status Code Messages

4510 This section defines status code enum keywords and values that are used to provide semantic information  
4511 on the results of an operation request. Each operation response MUST include a status code. ~~For error~~  
4512 ~~type status codes,~~ The response MAY also contain a status message that provides a short textual  
4513 description of the status. The status code is intended for use by automata, and the status message is  
4514 intended for the human end user. Since the status message is an OPTIONAL component of the operation  
4515 response, an IPP application (i.e., a browser, GUI, print driver or gateway) is NOT REQUIRED to  
4516 examine or display the status message, since it MAY not be returned to the application.

4517 The prefix of the status keyword defines the class of response as follows:

4518 "informational" - Request received, continuing process  
4519 "successful" - The action was successfully received, understood, and accepted  
4520 "redirection" - Further action must be taken in order to complete the request  
4521 "client-error" - The request contains bad syntax or cannot be fulfilled  
4522 "server-error" - The IPP object failed to fulfill an apparently valid request  
4523

4524 ~~As with type2 enums, Since~~ IPP status codes ~~are type2 enums, they~~ are extensible. IPP clients are NOT  
4525 REQUIRED to understand the meaning of all registered status codes, though such understanding is  
4526 obviously desirable. However, ~~applications-IPP clients~~ SHALL understand the class of any status code,  
4527 as indicated by the prefix, and treat any unrecognized response as being equivalent to the first status code  
4528 of that class, with the exception that an unrecognized response ~~shall not~~ **SHALL NOT** be cached. For  
4529 example, if an unrecognized status code of "client-error-xxx-yyy" is received by the client, it can safely  
4530 assume that there was something wrong with its request and treat the response as if it had received a  
4531 "client-error-bad-request" status code. In such cases, IPP applications SHOULD present the  
4532 OPTIONAL message (if present) to the end user since the message is likely to contain human readable  
4533 information which will help to explain the unusual status. The name of the enum is the suggested status  
4534 message for US English.

4535 The status code values range from 0x0000 to 0x7FFF. The value ranges for each status code class are as  
4536 follows:

4537 "successful" - 0x0000 to 0x00FF  
4538 "informational" - 0x0100 to 0x01FF  
4539 "redirection" - 0x0200 to 0x02FF  
4540 "client-error" - 0x0400 to 0x04FF  
4541 "server-error" - 0x0500 to 0x05FF  
4542

4543 The top half (128 values) of each range (0x0n40 to 0x0nFF, for n = 0 to 5) is reserved for private use  
4544 within each status code class. Values 0x0600 to 0x7FFF are reserved for future assignment and SHALL  
4545 ~~not~~ **NOT** be used.

#### 4546 ~~13.14.1~~ 14.1 Status Codes

4547 Each status code is described below. Section 14.2 contains a table that indicates which status codes apply  
4548 to which operations. Sections 16.3 and 16.4 describe the suggested steps for processing IPP attributes  
4549 for all operations, including returning status codes.

4550 ~~13.1.1~~14.1.1 Informational

4551 This class of status code indicates a provisional response and is to be used for informational purposes  
4552 only.

4553 There are no status codes defined in IPP/1.0 for this class of status code.

4554 ~~13.1.2~~14.1.2 Successful Status Codes

4555 This class of status code indicates that the client's request was successfully received, understood, and  
4556 accepted.

4557 ~~13.1.2.1~~14.1.2.1 successful-ok (0x0000)

4558 The request has succeeded. In the case of a ~~response to a Print-Jobcreate operation request~~, the  
4559 'successful-ok' status code indicates that the request was successfully received ~~and~~, validated, ~~processed~~,  
4560 and that the Job object has been created; it does not indicate that the job has been ~~printed~~processed. The  
4561 transition of the Job object into the 'completed' state is the only indicator that the job has been printed.

4562 ~~13.1.2.2~~14.1.2.2 successful-ok-ignored-or-substituted-attributes (0x0001)

4563 The request has succeeded, but some attributes were ignored or unsupported values were substituted  
4564 with supported values in order to process the job without rejecting it.

4565 ~~13.1.2.3~~14.1.2.3 successful-ok-conflicting-attributes (0x0002)

4566 The request has succeeded, but some attribute values conflicted with the values of other attributes. These  
4567 conflicting values were either (1) substituted with (supported) values or (2) the attributes were removed  
4568 in order to process the job without rejecting it.

4569 ~~13.1.3~~14.1.3 Redirection Status Codes

4570 This class of status code indicates that further action needs to be taken to fulfill the request.

4571 There are no status codes defined in IPP/1.0 for this class of status code.



4572 ~~13.1.4.1~~14.1.4 Client Error Status Codes

4573 This class of status code is intended for cases in which the client seems to have erred. The IPP object  
4574 SHOULD return a message containing an explanation of the error situation and whether it is a temporary  
4575 or permanent condition.

4576 ~~13.1.4.1~~14.1.4.1 client-error-bad-request (0x0400)

4577 The request could not be understood by the IPP object due to malformed syntax (such as the value of a  
4578 fixed length attribute whose length does not match the prescribed length for that attribute - see section  
4579 16.3). The IPP application SHOULD NOT repeat the request without modifications.

4580 ~~13.1.4.2~~14.1.4.2 client-error-forbidden (0x0401)

4581 The IPP object understood the request, but is refusing to fulfill it. Additional authentication information  
4582 or authorization credentials will not help and the request SHOULD NOT be repeated. This status code is  
4583 commonly used when the IPP object does not wish to reveal exactly why the request has been refused or  
4584 when no other response is applicable.

4585 ~~13.1.4.3~~14.1.4.3 client-error-not-authenticated (0x0402)

4586 The request requires user authentication. The IPP client may repeat the request with suitable  
4587 authentication information. If the request already included authentication information, then this status  
4588 code indicates that authorization has been refused for those credentials. If this response contains the  
4589 same challenge as the prior response, and the user agent has already attempted authentication at least  
4590 once, then the response message may contain relevant diagnostic information. This status codes reveals  
4591 more information than "client-error-forbidden".

4592 ~~13.1.4.4~~14.1.4.4 client-error-not-authorized (0x0403)

4593 The requester is not authorized to perform the request. Additional authentication information or  
4594 authorization credentials will not help and the request SHOULD NOT be repeated. This status code is  
4595 used when the IPP object wishes to reveal that the authentication information is understandable, however,  
4596 the requester is explicitly not authorized to perform the request. This status codes reveals more  
4597 information than "client-error-forbidden" and "client-error-not-authenticated".

4598 ~~13.1.4.5~~14.1.4.5 client-error-not-possible (0x0404)

4599 This status code is used when the request is for something that can not happen. For example, there might  
4600 be a request to cancel a job that has already been canceled or aborted by the system. The IPP client  
4601 SHOULD NOT repeat the request.

4602 ~~13.1.4.6~~14.1.4.6 client-error-timeout (0x0405)

4603 The client did not produce a request within the time that the IPP object was prepared to wait. For  
4604 example, a client issued a Create-Job operation and then, after a long period of time, issued a Send-  
4605 Document operation and this error status code was returned in response to the Send-Document request  
4606 (see section 3.3.1). The IPP object might have been forced to clean up resources that had been held for  
4607 the waiting additional Documents. The IPP object was forced to close the Job since the client took too  
4608 long. The client SHOULD NOT repeat the request without modifications.

4609 ~~13.1.4.7~~14.1.4.7 client-error-not-found (0x0406)

4610 The IPP object has not found anything matching the request URI. No indication is given of whether the  
4611 condition is temporary or permanent. For example, a client with an old reference to a Job (a URI) tries to  
4612 cancel the Job, however in the mean time the Job might have been completed and all record of it at the  
4613 Printer has been deleted. This status code, 'client-error-not-found' is returned indicating that the  
4614 referenced Job can not be found. This error status code is also used when a client supplies a URI as a  
4615 reference to the document data in either a Print-URI or Send-URI operation, but the document can not  
4616 be found.

4617 In practice, an IPP application should avoid a not found situation by first querying and presenting a list of  
4618 valid Printer URIs and Job URIs to the end-user.

4619 ~~13.1.4.8~~14.1.4.8 client-error-gone (0x0407)

4620 The requested object is no longer available and no forwarding address is known. This condition should  
4621 be considered permanent. Clients with link editing capabilities should delete references to the request  
4622 URI after user approval. If the IPP object does not know or has no facility to determine, whether or not  
4623 the condition is permanent, the status code "client-error-not-found" should be used instead.

4624 This response is primarily intended to assist the task of maintenance by notifying the recipient that the  
4625 resource is intentionally unavailable and that the IPP object administrator desires that remote links to that  
4626 resource be removed. It is not necessary to mark all permanently unavailable resources as "gone" or to  
4627 keep the mark for any length of time -- that is left to the discretion of the IPP object administrator.

4628 ~~13.1.4.9~~14.1.4.9 client-error-request-entity-too-large (0x0408)

4629 The IPP object is refusing to process a request because the request entity is larger than the IPP object is  
4630 willing or able to process. An IPP Printer returns this status code when it limits the size of print jobs and  
4631 it receives a print job that exceeds that limit or when the attributes are so many that their encoding causes  
4632 the request entity to exceed IPP object capacity.

4633 ~~13.1.4.10~~14.1.4.10 client-error-request-value-too-long (0x0409)

4634 The IPP object is refusing to service the request because one or more of the client client-supplied  
4635 attributes has a variable length value that is longer than ~~the IPP object is willing to interpret~~the maximum  
4636 length specified for that attribute. The IPP object might not have sufficient resources (memory, buffers,  
4637 etc.) to process (even temporarily), interpret, and/or ignore the large value. Another use of this error  
4638 code is when the IPP object supports the processing of the large value, but during the processing of the  
4639 request as a whole, the object may pass the value onto some other system component which is not able to  
4640 accept the large value. For more details, see section 16.3.

4641 Note: For attribute values that are URIs, this rare condition is only likely to occur when a client has  
4642 improperly submitted a request with long query information (e.g. an IPP application allows an end-user to  
4643 enter an invalid URI), when the client has descended into a URI "black hole" of redirection (e.g., a  
4644 redirected URI prefix that points to a suffix of itself), or when the IPP object is under attack by a client  
4645 attempting to exploit security holes present in some IPP objects using fixed-length buffers for reading or  
4646 manipulating the Request-URI.

4647 ~~13.1.4.11~~14.1.4.11 client-error-document-format-not-supported (0x040A)

4648 The IPP object is refusing to service the request because the document data is in a format, as specified in  
4649 the "document-format" operation attribute, that is not supported by the Printer object. This error is  
4650 returned independent of the client-supplied "ipp-attribute-fidelity". The Printer object SHALL return this  
4651 status code, even if there are other attributes that are not supported as well, since this error is a bigger  
4652 problem than with Job Template attributes.

4653 ~~13.1.4.12~~14.1.4.12 client-error-attributes-or-values-not-supported (0x040B)

4654 In a create request, if the Printer object does not support one or more attributes or attribute values  
4655 supplied in the request and the client supplied the "ipp-attributes-fidelity" operation attribute with the  
4656 'true' value, the Printer object shall return this status code. For example, if the request indicates 'iso-a4'  
4657 media, but that media type is not supported by the Printer object. Or, if the client supplies an optional  
4658 attribute and the attribute itself is not even supported by the Printer. If the "ipp-attribute-fidelity"  
4659 attribute is 'false', the Printer SHALL ignore or substitute values for unsupported attributes and values  
4660 rather than reject the request and return this status code.

4661 For any operation where a client requests attributes (such as a Get-Jobs, Get-Printer-Attributes, or Get-  
4662 Job-Attributes operation), if the IPP object does not support one or more of the requested attributes, the  
4663 IPP object simply ignores the unsupported requested attributes and processes the request as if they had  
4664 not been supplied, rather than returning this status code.

4665 ~~13.1.4.13~~14.1.4.13 client-error-uri-scheme-not-supported (0x040C)

4666 The type of the client supplied URI in a Print-URI or a Send-URI operation is not supported.

4667 ~~13.1.4.14~~14.1.4.14 client-error-charset-not-supported (0x040D)

4668 For any operation, if the IPP Printer does not support the charset supplied by the client in the "attributes-  
4669 charset" operation attribute, the Printer SHALL reject the operation and return this status (see Section  
4670 3.1.4.1).

4671 ~~13.1.4.15~~14.1.4.15 client-error-conflicting-attributes (0x040E)

4672 The request is rejected because some attribute values conflicted with the values of other attributes.

4673 ~~13.1.5.1~~14.1.5 Server Error Status Codes

4674 This class of status codes indicates cases in which the IPP object is aware that it has erred or is incapable  
4675 of performing the request. The IPP object SHOULD include a message containing an explanation of the  
4676 error situation, and whether it is a temporary or permanent condition.

4677 ~~13.1.5.1.1~~14.1.5.1 server-error-internal-error (0x0500)

4678 The IPP object encountered an unexpected condition that prevented it from fulfilling the request. This  
4679 error status code differs from "server-error-temporary-error" in that it implies a more permanent type of  
4680 internal error. It also differs from "server-error-device-error" in that it implies an unexpected condition  
4681 (unlike a paper-jam or out-of-toner problem which is undesirable but expected). This error status code  
4682 indicates that probably some knowledgeable human intervention is required.

4683 ~~13.1.5.2~~14.1.5.2 server-error-operation-not-supported (0x0501)

4684 The IPP object does not support the functionality required to fulfill the request. This is the appropriate  
4685 response when the IPP object does not recognize an operation or is not capable of supporting it.

4686 ~~13.1.5.3~~14.1.5.3 server-error-service-unavailable (0x0502)

4687 The IPP object is currently unable to handle the request due to a temporary overloading or maintenance  
4688 of the IPP object. The implication is that this is a temporary condition which will be alleviated after some  
4689 delay. If known, the length of the delay may be indicated in the message. If no delay is given, the IPP  
4690 application should handle the response as it would for a "server-error-temporary-error" response. If the  
4691 condition is more permanent, the error status codes "client-error-gone" or "client-error-not-found" could  
4692 be used.

4693 ~~13.1.5.4~~14.1.5.4 server-error-version-not-supported (0x0503)

4694 The IPP object does not support, or refuses to support, the IPP protocol version that was used in the  
4695 request message. The IPP object is indicating that it is unable or unwilling to complete the request using  
4696 the same version as supplied in the request other than with this error message. The response should  
4697 contain a Message describing why that version is not supported and what other versions are supported by  
4698 that IPP object.

4699 A conforming IPP/1.0 client SHALL specify the valid version ('1.0') on each request. A conforming  
4700 IPP/1.0 object SHALL NOT return this status code to a conforming IPP/1.0 client. An IPP object  
4701 SHALL return this status code to a non-conforming IPP client. The response SHALL identify in the  
4702 "version-number" operation attribute the closest version number that the IPP object does support.

4703 ~~13.1.5.5~~14.1.5.5 server-error-device-error (0x0504)

4704 A printer error, such as a paper jam, occurs while the IPP object processes a Print or Send operation.  
4705 The response contains the true Job Status (the values of the "job-state" and "job-state-reasons"  
4706 attributes). Additional information can be returned in the optional "job-state-message" attribute value or  
4707 in the OPTIONAL status message that describes the error in more detail. This error status code is only  
4708 returned in situations where the Printer is unable to accept the create request because of such a device  
4709 error. For example, if the Printer is unable to spool, and can only accept one job at a time, the reason it  
4710 might reject a create request is that the printer currently has a paper jam. In many cases however, where  
4711 the Printer object can accept the request even though the Printer has some error condition, the  
4712 'successful-ok' status code will be returned. In such a case, the client would look at the returned Job  
4713 Object Attributes or later query the Printer to determine its state and state reasons.

4714 ~~13.1.5.6~~14.1.5.6 server-error-temporary-error (0x0505)

4715 A temporary error such as a buffer full write error, a memory overflow (i.e. the document data exceeds  
4716 the memory of the Printer), or a disk full condition, occurs while the IPP Printer processes an operation.  
4717 The client MAY try the unmodified request again at some later point in time with an expectation that the  
4718 temporary internal error condition may have been cleared. Alternatively, as an implementation option, a

4719 Printer object MAY delay the response until the temporary condition is cleared so that no error is  
4720 returned.

4721 ~~13.1.5.7~~14.1.5.7 server-error-not-accepting-jobs (0x0506)

4722 A temporary error indicating that the Printer is not currently accepting jobs, because the administrator has  
4723 set the value of the Printer's "printer-is-not-accepting-jobs" attribute to 'false' (by means outside of  
4724 IPP/1.0).

4725 ~~13.1.5.8~~14.1.5.8 server-error-busy (0x0507)

4726 A temporary error indicating that the Printer is too busy processing jobs and/or other requests. The client  
4727 SHOULD try the unmodified request again at some later point in time with an expectation that the  
4728 temporary busy condition will have been cleared.

4729 ~~13.214.2~~ Status Codes for IPP Operations

4730 PJ = Print-Job, PU = Print-URI, CJ = Create-Job, SD = Send-Document  
 4731 SU = Send-URI, V = Validate-Job, GA = Get-Job-Attributes and  
 4732 Get-Printer-Attributes, GJ = Get-Jobs, C = Cancel-Job

4733		IPP Operations								
4734	IPP Status Keyword	PJ	PU	CJ	SD	SU	V	GA	GJ	C
4735	-----	--	--	--	--	--	--	--	--	--
4736	successful-ok	x	x	x	x	x	x	x	x	x
4737	successful-ok-ignored-or-substituted-	x	x	x	x	x	x	x	x	x
4738	attributes									
4739	successful-ok-conflicting-attributes	x	x	x	x	x	x	x	x	x
4740	client-error-bad-request	x	x	x	x	x	x	x	x	x
4741	client-error-forbidden	x	x	x	x	x	x	x	x	x
4742	client-error-not-authenticated	x	x	x	x	x	x	x	x	x
4743	client-error-not-authorized	x	x	x	x	x	x	x	x	x
4744	client-error-not-possible	x	x	x	x	x	x	x	x	x
4745	client-error-timeout	x	x	x	x	x	x	x	x	x
4746	client-error-not-found	x	x	x	x	x	x	x	x	x
4747	client-error-gone	x	x	x	x	x	x	x	x	x
4748	client-error-request-entity-too-large	x	x	x	x	x	x	x	x	x
4749	client-error-request- <del>value</del> -too-long	x	x	x	x	x	x	x	x	x
4750	client-error-document-format-not-	x	x		x	x	x	x		
4751	supported									
4752	client-error-attributes-or-values-not-	x	x	x	x	x	x	x	x	x
4753	supported									
4754	client-error-uri-scheme-not-supported		x			x				
4755	client-error-charset-not-supported	x	x	x	x	x	x	x	x	x
4756	client-error-conflicting-attributes	x	x	x	x	x	x	x	x	x
4757	server-error-internal-error	x	x	x	x	x	x	x	x	x
4758	server-error-operation-not-supported		x	x	x	x				
4759	server-error-service-unavailable	x	x	x	x	x	x	x	x	x
4760	server-error-version-not-supported	x	x	x	x	x	x	x	x	x
4761	server-error-device-error	x	x	x	x	x				
4762	server-error-temporary-error	x	x	x	x	x				
4763	server-error-not-accepting-jobs	x	x	x	x	x	x			
4764	server-error-busy	x	x	x	x	x	x	x	x	x
4765										
4766										
4767										

4768 ~~14.15.~~ APPENDIX C: "media" keyword values

4769 Standard keyword values are taken from several sources.

4770 Standard values are defined (taken from DPA[ISO10175] and the Printer MIB[RFC1759]):

4771 'default': The default medium for the output device  
4772 'iso-a4-white': Specifies the ISO A4 white medium  
4773 'iso-a4-colored': Specifies the ISO A4 colored medium  
4774 'iso-a4-transparent' Specifies the ISO A4 transparent medium  
4775 'iso-a3-white': Specifies the ISO A3 white medium  
4776 'iso-a3-colored': Specifies the ISO A3 colored medium  
4777 'iso-a5-white': Specifies the ISO A5 white medium  
4778 'iso-a5-colored': Specifies the ISO A5 colored medium  
4779 'iso-b4-white': Specifies the ISO B4 white medium  
4780 'iso-b4-colored': Specifies the ISO B4 colored medium  
4781 'iso-b5-white': Specifies the ISO B5 white medium  
4782 'iso-b5-colored': Specifies the ISO B5 colored medium  
4783 'jis-b4-white': Specifies the JIS B4 white medium  
4784 'jis-b4-colored': Specifies the JIS B4 colored medium  
4785 'jis-b5-white': Specifies the JIS B5 white medium  
4786 'jis-b5-colored': Specifies the JIS B5 colored medium  
4787

4788 The following standard values are defined for North American media:

4789 'na-letter-white': Specifies the North American letter white medium  
4790 'na-letter-colored': Specifies the North American letter colored medium  
4791 'na-letter-transparent': Specifies the North American letter transparent medium  
4792 'na-legal-white': Specifies the North American legal white medium  
4793 'na-legal-colored': Specifies the North American legal colored medium  
4794

4795 The following standard values are defined for envelopes:

4796 'iso-b4-envelope': Specifies the ISO B4 envelope medium  
4797 'iso-b5-envelope': Specifies the ISO B5 envelope medium  
4798 'iso-c3-envelope': Specifies the ISO C3 envelope medium  
4799 'iso-c4-envelope': Specifies the ISO C4 envelope medium  
4800 'iso-c5-envelope': Specifies the ISO C5 envelope medium  
4801 'iso-c6-envelope': Specifies the ISO C6 envelope medium  
4802 'iso-designated-long-envelope': Specifies the ISO Designated Long envelope medium  
4803 'na-10x13-envelope': Specifies the North American 10x13 envelope medium  
4804 'na-9x12-envelope': Specifies the North American 9x12 envelope medium  
4805 'monarch-envelope': Specifies the Monarch envelope  
4806 'na-number-10-envelope': Specifies the North American number 10 business envelope medium



4807 'na-7x9-envelope': Specifies the North American 7x9 inch envelope  
4808 'na-9x11-envelope': Specifies the North American 9x11 inch envelope  
4809 'na-10x14-envelope': Specifies the North American 10x14 inch envelope  
4810 'na-number-9-envelope': Specifies the North American number 9 business envelope  
4811 'na-6x9-envelope': Specifies the North American 6x9 inch envelope  
4812 'na-10x15-envelope': Specifies the North American 10x15 inch envelope  
4813

4814 The following standard values are defined for the less commonly used media (white-only):

4815 'executive-white': Specifies the white executive medium  
4816 'folio-white': Specifies the folio white medium  
4817 'invoice-white': Specifies the white invoice medium  
4818 'ledger-white': Specifies the white ledger medium  
4819 'quarto-white': Specifies the white quarto medium  
4820 'iso-a0-white': Specifies the ISO A0 white medium  
4821 'iso-a1-white': Specifies the ISO A1 white medium  
4822 'iso-a2-white': Specifies the ISO A2 white medium  
4823 'iso-a6-white': Specifies the ISO A6 white medium  
4824 'iso-a7-white': Specifies the ISO A7 white medium  
4825 'iso-a8-white': Specifies the ISO A8 white medium  
4826 'iso-a9-white': Specifies the ISO A9 white medium  
4827 'iso-10-white': Specifies the ISO A10 white medium  
4828 'iso-b0-white': Specifies the ISO B0 white medium  
4829 'iso-b1-white': Specifies the ISO B1 white medium  
4830 'iso-b2-white': Specifies the ISO B2 white medium  
4831 'iso-b3-white': Specifies the ISO B3 white medium  
4832 'iso-b6-white': Specifies the ISO B6 white medium  
4833 'iso-b7-white': Specifies the ISO B7 white medium  
4834 'iso-b8-white': Specifies the ISO B8 white medium  
4835 'iso-b9-white': Specifies the ISO B9 white medium  
4836 'iso-b10-white': Specifies the ISO B10 white medium  
4837 'jis-b0-white': Specifies the JIS B0 white medium  
4838 'jis-b1-white': Specifies the JIS B1 white medium  
4839 'jis-b2-white': Specifies the JIS B2 white medium  
4840 'jis-b3-white': Specifies the JIS B3 white medium  
4841 'jis-b6-white': Specifies the JIS B6 white medium  
4842 'jis-b7-white': Specifies the JIS B7 white medium  
4843 'jis-b8-white': Specifies the JIS B8 white medium  
4844 'jis-b9-white': Specifies the JIS B9 white medium  
4845 'jis-b10-white': Specifies the JIS B10 white medium

4846

4847 The following standard values are defined for engineering media:

4848 'a': Specifies the engineering A size medium

4849 'b': Specifies the engineering B size medium

4850 'c': Specifies the engineering C size medium

4851 'd': Specifies the engineering D size medium

4852 'e': Specifies the engineering E size medium

4853

4854 The following standard values are defined for input-trays (from ISO DPA and the Printer MIB):

4855 'top': The top input tray in the printer.

4856 'middle': The middle input tray in the printer.

4857 'bottom': The bottom input tray in the printer.

4858 'envelope': The envelope input tray in the printer.

4859 'manual': The manual feed input tray in the printer.

4860 'large-capacity': The large capacity input tray in the printer.

4861 'main': The main input tray

4862 'side': The side input tray

4863

4864 The following standard values are defined for media sizes (from ISO DPA):

4865 'iso-a0': Specifies the ISO A0 size: 841 mm by 1189 mm as defined in ISO 216

4866 'iso-a1': Specifies the ISO A1 size: 594 mm by 841 mm as defined in ISO 216

4867 'iso-a2': Specifies the ISO A2 size: 420 mm by 594 mm as defined in ISO 216

4868 'iso-a3': Specifies the ISO A3 size: 297 mm by 420 mm as defined in ISO 216

4869 'iso-a4': Specifies the ISO A4 size: 210 mm by 297 mm as defined in ISO 216

4870 'iso-a5': Specifies the ISO A5 size: 148 mm by 210 mm as defined in ISO 216

4871 'iso-a6': Specifies the ISO A6 size: 105 mm by 148 mm as defined in ISO 216

4872 'iso-a7': Specifies the ISO A7 size: 74 mm by 105 mm as defined in ISO 216

4873 'iso-a8': Specifies the ISO A8 size: 52 mm by 74 mm as defined in ISO 216

4874 'iso-a9': Specifies the ISO A9 size: 37 mm by 52 mm as defined in ISO 216

4875 'iso-a10': Specifies the ISO A10 size: 26 mm by 37 mm as defined in ISO 216

4876 'iso-b0': Specifies the ISO B0 size: 1000 mm by 1414 mm as defined in ISO 216

4877 'iso-b1': Specifies the ISO B1 size: 707 mm by 1000 mm as defined in ISO 216

4878 'iso-b2': Specifies the ISO B2 size: 500 mm by 707 mm as defined in ISO 216

4879 'iso-b3': Specifies the ISO B3 size: 353 mm by 500 mm as defined in ISO 216

4880 'iso-b4': Specifies the ISO B4 size: 250 mm by 353 mm as defined in ISO 216

4881 'iso-b5': Specifies the ISO B5 size: 176 mm by 250 mm as defined in ISO 216

4882 'iso-b6': Specifies the ISO B6 size: 125 mm by 176 mm as defined in ISO 216  
4883 'iso-b7': Specifies the ISO B7 size: 88 mm by 125 mm as defined in ISO 216  
4884 'iso-b8': Specifies the ISO B8 size: 62 mm by 88 mm as defined in ISO 216  
4885 'iso-b9': Specifies the ISO B9 size: 44 mm by 62 mm as defined in ISO 216  
4886 'iso-b10': Specifies the ISO B10 size: 31 mm by 44 mm as defined in ISO 216  
4887 'na-letter': Specifies the North American letter size: 8.5 inches by 11 inches  
4888 'na-legal': Specifies the North American legal size: 8.5 inches by 14 inches  
4889 'executive': Specifies the executive size (7.25 X 10.5 in)  
4890 'folio': Specifies the folio size (8.5 X 13 in)  
4891 'invoice': Specifies the invoice size (5.5 X 8.5 in)  
4892 'ledger': Specifies the ledger size (11 X 17 in)  
4893 'quarto': Specifies the quarto size (8.5 X 10.83 in)  
4894 'iso-c3': Specifies the ISO C3 size: 324 mm by 458 mm as defined in ISO 269  
4895 'iso-c4': Specifies the ISO C4 size: 229 mm by 324 mm as defined in ISO 269  
4896 'iso-c5': Specifies the ISO C5 size: 162 mm by 229 mm as defined in ISO 269  
4897 'iso-c6': Specifies the ISO C6 size: 114 mm by 162 mm as defined in ISO 269  
4898 'iso-designated-long': Specifies the ISO Designated Long size: 110 mm by 220 mm as defined in ISO  
4899 269  
4900 'na-10x13-envelope': Specifies the North American 10x13 size: 10 inches by 13 inches  
4901 'na-9x12-envelope': Specifies the North American 9x12 size: 9 inches by 12 inches  
4902 'na-number-10-envelope': Specifies the North American number 10 business envelope size: 4.125  
4903 inches by 9.5 inches  
4904 'na-7x9-envelope': Specifies the North American 7x9 inch envelope size  
4905 'na-9x11-envelope': Specifies the North American 9x11 inch envelope size  
4906 'na-10x14-envelope': Specifies the North American 10x14 inch envelope size  
4907 'na-number-9-envelope': Specifies the North American number 9 business envelope size  
4908 'na-6x9-envelope': Specifies the North American 6x9 envelope size  
4909 'na-10x15-envelope': Specifies the North American 10x15 envelope size  
4910 'monarch-envelope': Specifies the Monarch envelope size (3.87 x 7.5 in)  
4911 'jis-b0': Specifies the JIS B0 size: 1030mm x 1456mm  
4912 'jis-b1': Specifies the JIS B1 size: 728mm x 1030mm  
4913 'jis-b2': Specifies the JIS B2 size: 515mm x 728mm  
4914 'jis-b3': Specifies the JIS B3 size: 364mm x 515mm  
4915 'jis-b4': Specifies the JIS B4 size: 257mm x 364mm  
4916 'jis-b5': Specifies the JIS B5 size: 182mm x 257mm  
4917 'jis-b6': Specifies the JIS B6 size: 128mm x 182mm  
4918 'jis-b7': Specifies the JIS B7 size: 91mm x 128mm  
4919 'jis-b8': Specifies the JIS B8 size: 64mm x 91mm  
4920 'jis-b9': Specifies the JIS B9 size: 45mm x 64mm  
4921 'jis-b10': Specifies the JIS B10 size: 32mm x 45mm

4922 ~~15.16.~~ APPENDIX D: Processing IPP Attributes

4923 When submitting a print job to a Printer object, the IPP model allows a client to supply operation and Job  
4924 Template attributes along with the document data. These Job Template attributes in the create request  
4925 affect the rendering, production and finishing of the documents in the job. Similar types of instructions  
4926 may also be contained in the document to be printed, that is, embedded within the print data itself. In  
4927 addition, the Printer has a set of attributes that describe what rendering and finishing options which are  
4928 supported by that Printer. This model, which allows for flexibility and power, also introduces the  
4929 potential that at job submission time, these client-supplied attributes may conflict with either:

- 4930 - what the implementation is capable of realizing (i.e., what the Printer supports), as well as
- 4931 - the instructions embedded within the print data itself.

4932  
4933 The following sections describe how these two types of conflicts are handled in the IPP model.

4934 ~~15.16.1~~ Fidelity

4935 If there is a conflict between what the client requests and what a Printer object supports, the client may  
4936 request one of two possible conflict handling mechanisms:

- 4937 1) either reject the job since the job can not be processed exactly as specified, or
- 4938 2) allow the Printer to make any changes necessary to proceed with processing the Job the best it can.

4939  
4940 In the first case the client is indicating to the Printer object: "Print the job exactly as specified with no  
4941 exceptions, and if that can't be done, don't even bother printing the job at all." In the second case, the  
4942 client is indicating to the Printer object: "It is more important to make sure the job is printed rather than  
4943 be processed exactly as specified; just make sure the job is printed even if client supplied attributes need  
4944 to be changed or ignored."

4945 The IPP model accounts for this situation by introducing an "ipp-attribute-fidelity" attribute.

4946 In a create request, "ipp-attribute-fidelity" is a boolean operation attribute that is **OPTIONALLY**  
4947 supplied by the client. The value 'true' indicates that total fidelity to client supplied Job Template  
4948 attributes and values is required. The client is requesting that the Job be printed exactly as specified, and  
4949 if that is not possible then the job **MUST** be rejected rather than processed incorrectly. The value 'false'  
4950 indicates that a reasonable attempt to print the Job is acceptable. If a Printer does not support some of  
4951 the client supplied Job Template attributes or values, the Printer **SHALL** ignore them or substitute any  
4952 supported value for unsupported values, respectively. The Printer may choose to substitute the default  
4953 value associated with that attribute, or use some other supported value that is similar to the unsupported  
4954 requested value. For example, if a client supplies a "media" value of 'na-letter', the Printer may choose to

4955 substitute 'iso-a4' rather than a default value of 'envelope'. If the client does not supply the "ipp-attribute-  
4956 fidelity" attribute, the Printer assumes a value of 'false'.

4957 Each Printer implementation MUST support both types of "fidelity" printing (that is whether the client  
4958 supplies a value of 'true' or 'false'):

- 4959 - If the client supplies 'false' or does not supply the attribute, the Printer object SHALL always accept  
4960 the request by ignoring unsupported Job Template attributes and by substituting unsupported  
4961 values of supported Job Template attributes with supported values.
- 4962 - If the client supplies 'true', the Printer object SHALL reject the request if the client supplies  
4963 unsupported Job Template attributes.

4964

4965 Since a client can always query a Printer to find out exactly what is and is not supported, "ipp-attribute-  
4966 fidelity" set to 'false' is useful when:

- 4967 1) The End-User uses a command line interface to request attributes that might not be supported.
- 4968 2) In a GUI context, if the End User expects the job might be moved to another printer and prefers a  
4969 sub-optimal result to nothing at all.
- 4970 3) The End User just wants something reasonable in lieu of nothing at all.

4971

#### 4972 ~~4.5.2~~16.2 Page Description Language (PDL) Override

4973 If there is a conflict between the value of an IPP Job Template attribute and a corresponding instruction  
4974 in the document data, the value of the IPP attribute SHOULD take precedence over the document  
4975 instruction. Consider the case where a previously formatted file of document data is sent to an IPP  
4976 Printer. In this case, if the client supplies any attributes at job submission time, the client desires that  
4977 those attributes override the embedded instructions. Consider the case were a previously formatted  
4978 document has embedded in it commands to load 'iso-a4' media. However, the document is passed to an  
4979 end user that only has access to a printer with 'na-letter' media loaded. That end user most likely wants to  
4980 submit that document to an IPP Printer with the "media" Job Template attribute set to 'na-letter'. The job  
4981 submission attribute should take precedence over the embedded PDL instruction. However, until  
4982 companies that supply document data interpreters allow a way for external IPP attributes to take  
4983 precedence over embedded job production instructions, a Printer might not be able to support the  
4984 semantics that IPP attributes override the embedded instructions.

4985 The IPP model accounts for this situation by introducing a "pdl-override-supported" attribute that  
4986 describes the Printer objects capabilities to override instructions embedded in the PDL data stream. The  
4987 value of the "pdl-override-supported" attribute is configured by means outside IPP/1.0.

4988 This MANDATORY Printer attribute takes on the following values:

- 4989 - 'attempted': This value indicates that the Printer object attempts to make the IPP attribute values  
4990 take precedence over embedded instructions in the document data, however there is no guarantee.  
4991 - 'not-attempted': This value indicates that the Printer object makes no attempt to make the IPP  
4992 attribute values take precedence over embedded instructions in the document data.  
4993

4994 At job processing time, an implementation that supports the value of 'attempted' might do one of several  
4995 different actions:

- 4996 1) Generate an output device specific command sequence to realize the feature represented by the IPP  
4997 attribute value.
- 4998 2) Parse the document data itself and replace the conflicting embedded instruction with a new  
4999 embedded instruction that matches the intent of the IPP attribute value.
- 5000 3) Indicate to the Printer that external supplied attributes take precedence over embedded instructions  
5001 and then pass the external IPP attribute values to the document data interpreter.
- 5002 4) Anything else that allows for the semantics that IPP attributes override embedded document data  
5003 instructions.  
5004

5005 Since 'attempted' does not offer any type of guarantee, even though a given Printer object might not do a  
5006 very "good" job of attempting to ensure that IPP attributes take a higher precedence over instructions  
5007 embedded in the document data, it would still be a conforming implementation.

5008 At job processing time, an implementation that supports the value of 'not-attempted' might do one of the  
5009 following actions:

- 5010 1) Simply pre-pend the document data with the PDL instruction that corresponds to the client-  
5011 supplied PDL attribute, such that if the document data also has the same PDL instruction, it will  
5012 override what the Printer object pre-pended. In other words, this implementation is using the  
5013 same implementation semantics for the client-supplied IPP attributes as for the Printer object  
5014 defaults.
- 5015 2) Parse the document data and replace the conflicting embedded instruction with a new embedded  
5016 instruction that approximates, but does not match, the semantic intent of the IPP attribute value.  
5017

5018 Note: The "ipp-attribute-fidelity" attribute applies to the Printer's ability to either accept or reject other  
5019 unsupported Job Template attributes. In other words, if "ipp-attribute-fidelity" is set to 'true', a Job is  
5020 accepted if and only if the client supplied Job Template attributes and values are supported by the Printer.  
5021 Whether these attributes actually affect the processing of the Job when the document data contains  
5022 embedded instructions depends on the ability of the Printer to override the instructions embedded in the  
5023 document data with the semantics of the IPP attributes. If the document data attributes can be  
5024 overridden ("pdl-override-supported" set to 'attempted'), the Printer makes an attempt to use the IPP  
5025 attributes when processing the Job. If the document data attributes can not be overridden ("pdl-override-  
5026 supported" set to 'not-attempted'), the Printer makes no attempt to override the embedded document data

5027 instructions with the IPP attributes when processing the Job, and hence, the IPP attributes may fail to  
5028 affect the Job processing and output when the corresponding instruction is embedded in the document  
5029 data.

5030 ~~45.3~~16.3 Suggested Operation Processing Steps for All Operations

5031 When an IPP object receives a request, the IPP object either accepts or rejects the request. In order to  
5032 determine whether or not to accept or reject the request, the IPP object SHOULD execute the following  
5033 steps. The order of the steps may be rearranged and/or combined, including making one or multiple  
5034 passes over the request. Therefore, the error status codes returned may differ between implementations.  
5035 The next section contains the additional steps for the Print-Job, Validate-Job, Print-URI, Create-Job,  
5036 Send-Document, and Send-URI operations that create jobs, adds documents, and validates jobs.

5037 In the following, processing continues step by step until a "RETURNS the xxx status code ..." statement  
5038 is encountered. Error returns are indicated by the verb: "REJECTS". Since clients have difficulty getting  
5039 the status code before sending all of the document data in a Print-Job request, clients SHOULD use the  
5040 Validate-Job operation before sending large documents to be printed, in order to validate whether the IPP  
5041 Printer will accept the job or not.

5042 It is assumed that security authentication and authorization has already taken place at a lower layer.

5043 ~~45.3.1~~16.3.1 Validate version number

5044 Every request and every response contains the "version-number" attribute. The value of this attribute is  
5045 the major and minor version number of the syntax and semantics that the client and IPP object is using,  
5046 respectively. The "version-number" attribute remains in a fixed position across all future versions so that  
5047 all clients and IPP object that support future versions can determine which version is being used. The IPP  
5048 object checks to see if the major version number supplied in the request is supported. If not, the Printer  
5049 object REJECTS the request and RETURNS the 'server-error-version-not-supported' status code in the  
5050 response. The IPP object returns in the "version-number" response attribute the major and minor version  
5051 for the error response. Thus the client can learn at least one major and minor version that the IPP object  
5052 supports. The IPP object is encouraged to return the closest version number to the one supplied by the  
5053 client.

5054 The checking of the minor version number is implementation dependent, however if the client supplied  
5055 minor version is explicitly supported, the IPP object SHALL respond using that identical minor version  
5056 number. If the requested minor version is not supported (the requested minor version is either higher or  
5057 lower) than a supported minor version, the IPP object SHOULD return the closest supported minor  
5058 version.

5059 ~~15.3.2~~16.3.2 Validate operation identifier

5060 The Printer object checks to see if the "operation-id" attribute supplied by the client is supported as  
5061 indicated in the Printer object's "printer-operations-supported" attribute. If not, the Printer REJECTS the  
5062 request and returns the 'server-error-operation-not-supported' status code in the response.

5063 ~~15.3.3~~16.3.3 Validate the request identifier

5064 The Printer object checks to see if the "request-id" attribute supplied by the client is in range. If the value  
5065 is not between 1 and  $2^{*}31 - 1$  (inclusive), the Printer object REJECTS the request and returns the  
5066 'client-error-bad-request' status code in the response.

5067 Note: The "version-number", attribute, "operation-id", and the "request-id" attributes in the same fixed  
5068 octet positions in all versions of the protocol. These fields are validated before proceeding with the rest  
5069 of the validation.

5070 ~~15.3.4~~16.3.4 Validate attribute group and attribute presence and order

5071 The order of the following validation steps depends on implementation.

5072 ~~15.3.4.1~~16.3.4.1 Validate the presence and order of attribute groups

5073 Client requests and IPP object responses contain attribute groups that Section 3 requires to be present  
5074 and in a specified order. An IPP object verifies that the attribute groups are present and in the correct  
5075 order in requests supplied by clients (attribute groups without an \* in the following tables).

5076 If an IPP object receives a request with (1) required attribute groups missing, or (2) the attributes groups  
5077 are out of order, or (3) the groups are repeated, the IPP object REJECTS the request and RETURNS the  
5078 'client-error-bad-request' status code. For example, it is an error for the Job Template Attributes group  
5079 to occur before the Operation Attributes group, for the Operation Attributes group to be omitted, or for  
5080 an attribute group to occur more than once, except in the Get-Jobs response.

5081 Since this kind of attribute group error is most likely to be an error detected by a client developer rather  
5082 than by a customer, the IPP object NEED NOT return an indication of which attribute group was in error  
5083 in either the Unsupported Attributes group or the Status Message. Also, the IPP object NEED NOT find  
5084 all attribute group errors before returning this error.

5085 ~~15.3.4.2~~16.3.4.2 Ignore unknown attribute groups in the expected position

5086 Future attribute groups may be added to the specification at the end of requests just before the Document  
5087 Content and at the end of response, except for the Get-Jobs response, where it maybe there or before the



5088 first job attributes returned. If an IPP object receives an unknown attribute group in these positions, it  
5089 ignores the entire group, rather than returning an error, since that group may be a new group in a later  
5090 minor version of the protocol that can be ignored. (If the new attribute group cannot be ignored without  
5091 confusing the client, the major version number would have been increased in the protocol document and  
5092 in the request). If the unknown group occurs in a different position, the IPP object REJECTS the request  
5093 and RETURNS the 'client-error-bad-request' status code.

5094 Clients also ignore unknown attribute groups returned in a response.

5095 Note: By validating that requests are in the proper form, IPP objects force clients to use the proper form  
5096 which, in turn, increases the chances that customers will be able to use such clients from multiple vendors  
5097 with IPP objects from other vendors.

5098 ~~15.3.4.3~~16.3.4.3 Validate the presence of a single occurrence of required Operation attributes

5099 Client requests and IPP object responses contain Operation attributes that Section 3 requires to be  
5100 present. Attributes within a group may be in any order, except for the ordering of target, charset, and  
5101 natural languages attributes. These attributes must be first, and must be supplied in the following order:  
5102 charset, natural language, and then target. An IPP object verifies that the attributes that Section 4  
5103 requires to be supplied by the client have been supplied in the request (attributes without an \* in the  
5104 following tables). An asterisk (\*) indicates groups and Operation attributes that the client may omit in a  
5105 request or an IPP object may omit in a response.

5106 If an IPP object receives a request with required attributes missing or repeated from a group, the IPP  
5107 object REJECTS the request and RETURNS the 'client-error-bad-request' status code. For example, it is  
5108 an error for the "attributes-charset" or "attributes-natural-language" attribute to be omitted in any  
5109 operation request, or for an Operation attribute to be supplied in a Job Template group or a Job Template  
5110 attribute to be supplied in an Operation Attribute group in a create request. It is also an error to supply  
5111 the "attributes-charset" attribute twice.

5112 Since these kinds of attribute errors are most likely to be detected by a client developer rather than by a  
5113 customer, the IPP object NEED NOT return an indication of which attribute was in error in either the  
5114 Unsupported Attributes group or the Status Message. Also, the IPP object NEED NOT find all attribute  
5115 errors before returning this error.

5116 The following tables list all the attributes for all the operations by attribute group in each request and  
5117 each response. The order of the groups is the order that the client supplies the groups as specified in  
5118 Section 3. The order of the attributes within a group is arbitrary, except as noted for some of the special  
5119 operation attributes (charset, natural language, and target). The tables below use the following notation:

5120 M indicates a MANDATORY attribute that an IPP object MUST support

5121 O ~~indicated~~ indicates an OPTIONAL attribute that an IPP object NEED NOT support  
 5122 \* indicates that a client MAY omit the attribute in a request and that an IPP object MAY  
 5123 omit the attribute in a response. The absence of an \* means that a client MUST  
 5124 supply the attribute in a request and an IPP object MUST supply the attribute in a  
 5125 response.  
 5126  
 5127

### Operation Requests

5128 The tables below show the attributes in their proper attribute groups for operation requests:

5129 Note: All operation requests contain the following common elements:  
 5130 version-number, operation-id, and request-id.

5131  
 5132 Print-Job Request:

5133 Group 1: Operation Attributes (M)  
 5134 attributes-charset (M)  
 5135 attributes-natural-language (M)  
 5136 printer-uri (M)  
 5137 requesting-user-name (M\*)  
 5138 job-name (M\*)  
 5139 ipp-attribute-fidelity (M\*)  
 5140 document-name (M\*)  
 5141 document-format (M\*)  
 5142 document-natural-language (O\*)  
 5143 compression (O\*)  
 5144 job-k-octets (O\*)  
 5145 job-impressions (O\*)  
 5146 job-media-sheets (O\*)  
 5147 Group 2: Job Template Attributes (M)  
 5148 <Job Template attributes> (O\*) (see Section 4.2)  
 5149 Group 3: Document Content (M)  
 5150 <document content>  
 5151

5152 Validate-Job Request:

5153 Group 1: Operation Attributes (M)  
 5154 attributes-charset (M)  
 5155 attributes-natural-language (M)  
 5156 printer-uri (M)  
 5157 requesting-user-name (M\*)  
 5158 job-name (M\*)  
 5159 ipp-attribute-fidelity (M\*)  
 5160 document-name (M\*)  
 5161 document-format (M\*)  
 5162 document-natural-language (O\*)

5163           compression (O\*)  
5164           job-k-octets (O\*)  
5165           job-impressions (O\*)  
5166           job-media-sheets (O\*)  
5167       Group 2: Job Template Attributes (M)  
5168           <Job Template attributes> (O\*) (see Section 4.2)  
5169  
5170   Create-Job Request:  
5171       Group 1: Operation Attributes (M)  
5172           attributes-charset (M)  
5173           attributes-natural-language (M)  
5174           printer-uri (M)  
5175           requesting-user-name (M\*)  
5176           job-name (M\*)  
5177           ipp-attribute-fidelity (M\*)  
5178           job-k-octets (O\*)  
5179           job-impressions (O\*)  
5180           job-media-sheets (O\*)  
5181       Group 2: Job Template Attributes (M)  
5182           <Job Template attributes> (O\*) (see Section 4.2)  
5183  
5184   Print-URI Request:  
5185       Group 1: Operation Attributes (M)  
5186           attributes-charset (M)  
5187           attributes-natural-language (M)  
5188           printer-uri (M)  
5189           document-uri (M)  
5190           requesting-user-name (M\*)  
5191           job-name (M\*)  
5192           ipp-attribute-fidelity (M\*)  
5193           document-name (M\*)  
5194           document-format (M\*)  
5195           document-natural-language (O\*)  
5196           compression (O\*)  
5197           job-k-octets (O\*)  
5198           job-impressions (O\*)  
5199           job-media-sheets (O\*)  
5200       Group 2: Job Template Attributes (M)  
5201           <Job Template attributes> (O\*) (see Section 4.2)  
5202  
5203   Send-Document Request:  
5204       Group 1: Operation Attributes (M)  
5205           attributes-charset (M)  
5206           attributes-natural-language (M)  
5207           (printer-uri & job-id) | job-uri (M)

5208 last-document (M)  
5209 requesting-user-name (M\*)  
5210 document-name (M\*)  
5211 document-format (M\*)  
5212 document-natural-language (O\*)  
5213 compression (O\*)  
5214 Group 2: Document Content (M)  
5215 <document content>  
5216  
5217 Send-URI Request:  
5218 Group 1: Operation Attributes (M)  
5219 attributes-charset (M)  
5220 attributes-natural-language (M)  
5221 (printer-uri & job-id) | job-uri (M)  
5222 last-document (M)  
5223 document-uri (M)  
5224 requesting-user-name (M\*)  
5225 document-name (M\*)  
5226 document-format (M\*)  
5227 document-natural-language (O\*)  
5228 compression (O\*)  
5229  
5230 Cancel-Job Request:  
5231 Group 1: Operation Attributes (M)  
5232 attributes-charset (M)  
5233 attributes-natural-language (M)  
5234 (printer-uri & job-id) | job-uri (M)  
5235 requesting-user-name (M\*)  
5236 message (O\*)  
5237  
5238 Get-Printer-Attributes Request:  
5239 Group 1: Operation Attributes (M)  
5240 attributes-charset (M)  
5241 attributes-natural-language (M)  
5242 printer-uri (M)  
5243 requesting-user-name (M\*)  
5244 requested-attributes (M\*)  
5245 document-format (M\*)  
5246  
5247 Get-Job-Attributes Request:  
5248 Group 1: Operation Attributes (M)  
5249 attributes-charset (M)  
5250 attributes-natural-language (M)  
5251 (printer-uri & job-id) | job-uri (M)  
5252 requesting-user-name (M\*)



5294           attributes-natural-language (M)  
5295           status-message (O\*)  
5296       Group 2: Unsupported Attributes (M\*) (see Note 3)  
5297           <unsupported attributes> (M\*)  
5298

5299 Note 2 - the Job Object Attributes and Printer Object Attributes are  
5300 returned only if the IPP object returns one of the success status  
5301 codes.  
5302

5303 Note 3 - the Unsupported Attributes Group is present only if the  
5304 client included some Operation and/or Job Template attributes that the  
5305 Printer doesn't support whether a success or an error return.  
5306

5307 Get-Printer-Attributes Response:

5308       Group 1: Operation Attributes (M)  
5309           attributes-charset (M)  
5310           attributes-natural-language (M)  
5311           status-message (O\*)  
5312       Group 2: Unsupported Attributes (M\*) (see Note 4)  
5313           <unsupported attributes> (M\*)  
5314       Group 3: Printer Object Attributes(M\*) (see Note 2)  
5315           <requested attributes> (M\*)  
5316

5317 Note 4 - the Unsupported Attributes Group is present only if the  
5318 client included some Operation attributes that the Printer doesn't  
5319 support whether a success or an error return.  
5320

5321 Get-Job-Attributes Response:

5322       Group 1: Operation Attributes (M)  
5323           attributes-charset (M)  
5324           attributes-natural-language (M)  
5325           status-message (O\*)  
5326       Group 2: Unsupported Attributes (M\*) (see Note 4)  
5327           <unsupported attributes> (M\*)  
5328       Group 3: Job Object Attributes(M\*) (see Note 2)  
5329           <requested attributes> (M\*)  
5330

5331 Get-Jobs Response:

5332       Group 1: Operation Attributes (M)  
5333           attributes-charset (M)  
5334           attributes-natural-language (M)  
5335           status-message (O\*)  
5336       Group 2: Unsupported Attributes (M\*) (see Note 4)  
5337           <unsupported attributes> (M\*)  
5338       Group 3: Job Object Attributes(M\*) (see Note 2, 5)

5339 <requested attributes> (M\*)

5340

5341 Note 5: for the Get-Jobs operation the response contains a separate  
5342 Job Object Attributes group 3 to N containing requested-attributes for  
5343 each job object in the response.

5344

5345 ~~15.3.5~~16.3.5 Validate the values of the MANDATORY Operation attributes

5346 An IPP object validates the values supplied by the client of the MANDATORY Operation attribute that  
5347 the IPP object MUST support. The next section specifies the validation of the values of the OPTIONAL  
5348 Operation attributes that IPP objects MAY support.

5349 The IPP object performs the following syntactic validation checks of each Operation attribute value:

5350

- 5351 a) that the length of each Operation attribute value is correct for the attribute syntax tag supplied  
5352 by the client according to Section 4.1.
- 5353 b) that the attribute syntax tag is correct for that Operation attribute according to Section 3,  
5354 c) that the value is in the range specified for that Operation attribute according to Section 3,  
5355 d) that multiple values are supplied by the client only for operation attributes that are multi-  
5356 valued, i.e., that are 1setOf X according to Section 3.

5357

5358 If any of these checks fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-  
5359 request' ~~or the 'client-error-request-value-too-long'~~ status code. Since such an error is most likely to be  
5360 an error detected by a client developer, rather than by an end-user, the IPP object NEED NOT return an  
5361 indication of which attribute had the error in either the Unsupported Attributes Group or the Status  
5362 Message. The description for each of these syntactic checks is explicitly expressed in the first IF  
5363 statement in the following table.

5364 In addition, the IPP object checks each Operation attribute value against some Printer object attribute or  
5365 some hard-coded value if there is no "xxx-supported" Printer object attribute defined. If its value is not  
5366 among those supported or is not in the range supported, then the IPP object REJECTS the request and  
5367 RETURNS the error status code indicated in the table by the second IF statement. If the value of the  
5368 Printer object's "xxx-supported" attribute is 'no-value' (because the system administrator hasn't configured  
5369 a value), the check always fails.

5370 -----

5371 attributes-charset (charset)

5372 IF NOT any single non-empty 'charset' value less than ~~or equal to 64-63~~ octets, REJECT/RETURN  
5373 'client-error-~~bad-request~~request-value-too-long'.

5374 IF NOT in the Printer object's "charset-supported" attribute, REJECT/RETURN "client-error-  
5375 charset-not-supported".

5376  
5377 attributes-natural-language(naturalLanguage)  
5378 IF NOT any single non-empty 'naturalLanguage' value less than or equal to 64-63 octets,  
5379 REJECT/RETURN 'client-error-~~bad-request~~request-value-too-long'.  
5380 ACCEPT the request even if not a member of the set in the Printer object's "generated-natural-  
5381 language-supported" attribute.  
5382  
5383 requesting-user-name  
5384 IF NOT any single ~~non-empty~~ 'name' value less than or equal to 256-255 octets, REJECT/RETURN  
5385 'client-error-~~bad-request~~request-value-too-long'.  
5386 IF the IPP object can obtain a better authenticated name, use it instead.  
5387  
5388 job-name(name)  
5389 IF NOT any single ~~non-empty~~ 'name' value less than or equal to 256-255 octets, REJECT/RETURN  
5390 'client-error-~~bad-request~~request-value-too-long'.  
5391 IF NOT supplied by the client, the Printer object creates a name from the document-name or  
5392 document-uri.  
5393  
5394 document-name (name)  
5395 IF NOT any single ~~non-empty~~ 'name' value less than or equal to 256-255 octets, REJECT/RETURN  
5396 'client-error-~~bad-request~~request-value-too-long'.  
5397  
5398 ipp-attribute-fidelity (boolean)  
5399 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-  
5400 error-bad-request'.  
5401 IF NOT supplied by the client, the IPP object assumes the value 'false'.  
5402  
5403 document-format (mimeMediaType)  
5404 IF NOT any single non-empty 'mimeMediaType' value less than or equal to 64-255 octets,  
5405 REJECT/RETURN 'client-error-~~bad-request~~request-value-too-long'.  
5406 IF NOT in the Printer object's "document-format-supported" attribute, REJECT/RETURN 'client-  
5407 error-document-format-not-supported'.  
5408 IF NOT supplied by the client, the IPP object assumes the value of the Printer object's "document-  
5409 format-default" attribute.  
5410  
5411 document-uri (uri)  
5412 IF NOT any single non-empty 'uri' value less than or equal to 1024-1023 octets, REJECT/RETURN  
5413 'client-error-request-~~uri~~value-too-long'.  
5414 IF the URI syntax is not valid, REJECT/RETURN 'client-error-bad-request'.



5415 IF scheme is NOT in the Printer object's "reference-uri-schemes-supported" attribute,  
5416 REJECT/RETURN 'client-error'-uri-scheme-not-supported'.  
5417

5418 last-document (boolean)  
5419 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-  
5420 error-bad-request'.  
5421

5422 job-id (integer(1:MAX))  
5423 IF NOT any single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN  
5424 'client-error-bad-request'.  
5425 IF NOT a job-id of an existing Job object, REJECT/RETURN 'client-error-not-found' or 'client-error-  
5426 gone' status code, if keep track of recently deleted jobs.  
5427

5428 requested-attributes (1setOf keyword)  
5429 IF NOT any number of 'keyword' values less than or equal to 256-255 octets, REJECT/RETURN  
5430 'client-error-~~bad-request~~request-value-too-long'.  
5431 Ignore unsupported values which are the keyword names of unsupported attributes. Don't bother to  
5432 copy such requested (unsupported) attributes to the Unsupported Attribute response group since  
5433 the response will not return them.  
5434

5435 which-jobs (type2 keyword)  
5436 IF NOT a single 'keyword' value less than or equal to 256-255 octets, REJECT/RETURN 'client-  
5437 error-~~bad-request~~request-value-too-long'.  
5438 IF NEITHER 'completed' NOR 'not-completed', copy the attribute and the unsupported value to the  
5439 Unsupported Attributes response group and REJECT/RETURN 'client-error-attributes-or-values-  
5440 not-supported'.  
5441 Note: a Printer still supports the 'completed' value even if it keeps no completed/canceled/aborted  
5442 jobs: by returning no jobs when so queried.  
5443 IF NOT supplied by the client, the IPP object assumes the 'not-completed' value.  
5444

5445 my-jobs (boolean)  
5446 IF NOT either a single 'true' or 'false' 'boolean' value equal to 1 octet, REJECT/RETURN 'client-  
5447 error-bad-request'.  
5448 IF NOT supplied by the client, the IPP object assumes the 'false' value.  
5449

5450 limit (integer(1:MAX))  
5451 IF NOT any single 'integer' value equal to 4 octets AND in the range 1 to MAX, REJECT/RETURN  
5452 'client-error-bad-request'.  
5453 IF NOT supplied by the client, the IPP object returns all jobs, no matter how many.  
5454

5455 -----

5456

5457 ~~15.3.6~~16.3.6 Validate the values of the OPTIONAL Operation attributes

5458 OPTIONAL Operation attributes are those that an IPP object MAY or MAY NOT support. An IPP  
5459 object validates the values of the OPTIONAL attributes supplied by the client. The IPP object performs  
5460 the same syntactic validation checks for each OPTIONAL attribute value as in Section 16.3.5. As in  
5461 Section 16.3.5, if any fail, the IPP object REJECTS the request and RETURNS the 'client-error-bad-  
5462 request' or the 'client-error-request-value-too-long' status code.

5463 In addition, the IPP object checks each Operation attribute value against some Printer attribute or some  
5464 hard-coded value if there is no "xxx-supported" Printer attribute defined. If its value is not among those  
5465 supported or is not in the range supported, then the IPP object REJECTS the request and RETURNS the  
5466 error status code indicated in the table. If the value of the Printer object's "xxx-supported" attribute is  
5467 'no-value' (because the system administrator hasn't configured a value), the check always fails.

5468 If the IPP object doesn't recognize/support an attribute, the IPP object treats the attribute as an unknown  
5469 or unsupported attribute (see the last row in the table below).

5470 -----

5471 document-natural-language (naturalLanguage)

5472 IF NOT any single non-empty 'naturalLanguage' value less than or equal to 64-63 octets,  
5473 REJECT/RETURN 'client-error-~~bad-request~~request-value-too-long'.

5474 IF NOT a value that the Printer object supports in document formats, (no standard "xxx-supported"  
5475 Printer attribute), REJECT/RETURN 'client-error-natural-language-not-supported'.

5476

5477 compression (type3 keyword)

5478 IF NOT any single 'keyword' values less than or equal to 256-255 octets, REJECT/RETURN 'client-  
5479 error-~~bad-request~~request-value-too-long'.

5480 IF NOT in the Printer object's "compression-supported" attribute, copy the attribute and the  
5481 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-  
5482 error-attributes-or-values-not-supported'.

5483

5484 job-k-octets (integer(0:MAX))

5485 IF NOT any single 'integer' value equal to 4 octets,  
5486 REJECT/RETURN 'client-error-bad-request'.

5487 IF NOT in the range of the Printer object's "job-k-octets-supported" attribute, copy the attribute and  
5488 the unsupported value to the Unsupported Attributes response group and REJECT/RETURN  
5489 'client-error-attributes-or-values-not-supported'.

5490

5491 job-impressions (integer(0:MAX))  
5492 IF NOT any single 'integer' value equal to 4 octets,  
5493 REJECT/RETURN 'client-error-bad-request'.  
5494 IF NOT in the range of the Printer object's "job-impressions-supported" attribute, copy the attribute  
5495 and the unsupported value to the Unsupported Attributes response group and REJECT/RETURN  
5496 'client-error-attributes-or-values-not-supported'.  
5497  
5498 job-media-sheets (integer(0:MAX))  
5499 IF NOT any single 'integer' value equal to 4 octets,  
5500 REJECT/RETURN 'client-error-bad-request'.  
5501 IF NOT in the range of the Printer object's "job-media-supported" attribute, copy the attribute and the  
5502 unsupported value to the Unsupported Attributes response group and REJECT/RETURN 'client-  
5503 error-attributes-or-values-not-supported'.  
5504  
5505 message (text(127))  
5506 IF NOT any single ~~non-empty~~'text' value less than or equal to 128-127 octets,  
5507 REJECT/RETURN 'client-error-~~bad-request~~request-value-too-long'.  
5508  
5509 unknown or unsupported attribute  
5510 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute  
5511 syntax, REJECT/RETURN 'client-error-~~bad-request~~request-value-too-long'.  
5512 ELSE copy the attribute and value to the Unsupported Attributes response group and change the  
5513 attribute value to the "out-of-band" 'unsupported' value, but otherwise ignore the attribute.  
5514  
5515 Note: Future Operation attributes may be added to the protocol specification that may occur  
5516 anywhere in the specified group. When the operation is otherwise successful, the IPP object returns  
5517 the 'successful-ok-ignored-or-substituted-attributes' status code. Ignoring unsupported Operation  
5518 attributes in all operations is analogous to the handling of unsupported Job Template attributes in the  
5519 create and Validate-Job operations when the client supplies the "ipp-attribute-fidelity" Operation  
5520 attribute with the 'false' value. This last rule is so that we can add OPTIONAL Operation attributes to  
5521 future versions of IPP so that older clients can inter-work with new IPP objects and newer clients can  
5522 inter-work with older IPP objects. (If the new attribute cannot be ignored without performing  
5523 unexpectedly, the major version number would have been increased in the protocol document and in  
5524 the request). This rule for Operation attributes is independent of the value of the "ipp-attribute-  
5525 fidelity" attribute. For example, if an IPP object doesn't support the OPTIONAL "job-k-octets"  
5526 attribute', the IPP object treats "job-k-octets" as an unknown attribute and only checks the length for  
5527 the 'integer' attribute syntax supplied by the client. If it is not four octets, the IPP object REJECTS  
5528 the request and RETURNS the 'client-error-bad-request' status code, else the IPP object copies the  
5529 attribute to the Unsupported Attribute response group, setting the value to the "out-of-band"  
5530 'unsupported' value, but otherwise ignores the attribute.

5531 -----

5532 ~~15.4.1~~16.4 Suggested Additional Processing Steps for Operations that Create/Validate Jobs and Add  
5533 Documents

5534 This section in combination with the previous section recommends the processing steps for the Print-Job,  
5535 Validate-Job, Print-URI, Create-Job, Send-Document, and Send-URI operations that IPP objects  
5536 SHOULD use. These are the operations that create jobs, validate a Print-Job request, and add  
5537 documents to a job.

5538 ~~15.4.1~~16.4.1 Default "ipp-attribute-fidelity" if not supplied

5539 The Printer object checks to see if the client supplied an "ipp-attribute-fidelity" Operation attribute. If the  
5540 attribute is not supplied by the client, the IPP object assumes that the value is 'false'.

5541 ~~15.4.2~~16.4.2 Check that the Printer object is accepting jobs

5542 If the value of the Printer object's "printer-is-accepting-jobs" is 'false', the Printer object REJECTS the  
5543 request and RETURNS the 'server-error-not-accepting-jobs' status code.

5544 ~~15.4.3~~16.4.3 Validate the values of the Job Template attributes

5545 An IPP object validates the values of all Job Template attribute supplied by the client. The IPP object  
5546 performs the analogous syntactic validation checks of each Job Template attribute value that it performs  
5547 for Operation attributes (see Section 16.3.5.):

- 5548 a) that the length of each value is correct for the attribute syntax tag supplied by the client  
5549 according to Section 4.1.  
5550 b) that the attribute syntax tag is correct for that attribute according to Sections 4.2 to 4.4,  
5551 c) that multiple values are supplied only for multi-valued attributes, i.e., that are 1setOf X  
5552 according to Sections 4.2 to 4.4

5553

5554 As in Section 16.3.5, if any of these syntactic checks fail, the IPP object REJECTS the request and  
5555 RETURNS the 'client-error-bad-request' or 'client-error-request-value-too-long' status code, independent  
5556 of the value of the "ipp-attribute-fidelity". Since such an error is most likely to be an error detected by a  
5557 client developer, rather than by an end-user, the IPP object NEED NOT return an indication of which  
5558 attribute had the error in either the Unsupported Attributes Group or the Status Message. The  
5559 description for each of these syntactic checks is explicitly expressed in the first IF statement in the  
5560 following table.

5561 In addition, the IPP object loops through all the client-supplied Job Template attributes, checking to see if  
 5562 the supplied attribute value(s) are supported or in the range supported, i.e., the value of the "xxx"  
 5563 attribute in the request is (1) a member of the set of values or is in the range of values of the Printer'  
 5564 objects "xxx-supported" attribute. If the value of the Printer object's "xxx-supported" attribute is 'no-  
 5565 value' (because the system administrator hasn't configured a value), the check always fails. If the check  
 5566 fails, the IPP object copies the attribute to the Unsupported Attributes response group with its  
 5567 unsupported value. If the attribute contains more than one value, each value is checked and each  
 5568 unsupported value is separately copied, while supported values are not copied. If an IPP object doesn't  
 5569 recognize/support a Job Template attribute, i.e., there is no corresponding Printer object "xxx-supported"  
 5570 attribute, the IPP object treats the attribute as an unknown or unsupported attribute (see the last row in  
 5571 the table below).

5572 If some Job Template attributes are supported for some document formats and not for others or the  
 5573 values are different for different document formats, the IPP object SHOULD take that into account in  
 5574 this validation using the value of the "document-format" supplied by the client (or defaulted to the value  
 5575 of the Printer's "document-format-default" attribute, if not supplied by the client). For example, if  
 5576 "number-up" is supported for the 'text/plain' document format, but not for the 'application/postscript'  
 5577 document format, the check SHOULD (though it NEED NOT) depend on the value of the "document-  
 5578 format" operation attribute. See "document-format" in section 3.2.1.1 and 3.2.5.1.

5579 Note: whether the request is accepted or rejected is determined by the value of the "ipp-attribute-fidelity"  
 5580 attribute in a subsequent step, so that all Job Template attribute supplied are examined and all  
 5581 unsupported attributes and/or values are copied to the Unsupported Attributes response group.

5582 -----

5583 job-priority (integer(1:100))

5584 IF NOT any single 'integer' value equal to 4 octets, REJECT/RETURN 'client-error-bad-request'.

5585 IF NOT supplied by the client, use the value of the Printer object's "job-priority-default" attribute at  
 5586 job submission time.

5587 IF NOT in the range 1 to 100, inclusive, copy the attribute and the unsupported value to the  
 5588 Unsupported Attributes response group.

5589 Map the value to the nearest supported value in the range 1:100 as specified by the number of  
 5590 discrete values indicated by the value of the Printer's "job-priority-supported" attribute. See the  
 5591 formula in Section 4.2.1.

5592

5593 job-hold-until (type3 keyword | name)

5594 IF NOT any single 'keyword' or 'name' value less than or equal to 256-255 octets, REJECT/RETURN  
 5595 'client-error-~~bad-request~~request-value-too-long'.

5596 IF NOT supplied by the client, use the value of the Printer object's "job-hold-until" attribute at job  
 5597 submission time.

5598 IF NOT in the Printer object's "job-hold-until-supported" attribute, copy the attribute and the  
5599 unsupported value to the Unsupported Attributes response group.  
5600

5601 job-sheets (type3 keyword | name)  
5602 IF NOT any single 'keyword' or 'name' value less than or equal to 256-255 octets, REJECT/RETURN  
5603 'client-error-~~bad-request~~request-value-too-long'.  
5604 IF NOT in the Printer object's "job-sheets-supported" attribute, copy the attribute and the  
5605 unsupported value to the Unsupported Attributes response group.  
5606

5607 multiple-document-handling (type2 keyword)  
5608 IF NOT any single 'keyword' value less than or equal to 256-255 octets, REJECT/RETURN 'client-  
5609 error-~~bad-request~~request-value-too-long'.  
5610 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute  
5611 and the unsupported value to the Unsupported Attributes response group.  
5612

5613 copies (integer(1:MAX))  
5614 IF NOT any single 'integer' value equal to 4 octets,  
5615 REJECT/RETURN 'client-error-bad-request'.  
5616 IF NOT in range of the Printer object's "copies-supported" attribute  
5617 copy the attribute and the unsupported value to the Unsupported Attributes response group.  
5618

5619 finishings (1setOf type2 enum)  
5620 IF NOT any ~~'keyword' or 'nameenum'~~ value(s) equal to each less than 2564 octets,  
5621 REJECT/RETURN 'client-error-bad-request'.  
5622 IF NOT in the Printer object's "finishings-supported" attribute, copy the attribute and the unsupported  
5623 value(s), but not any supported values, to the Unsupported Attributes response group.  
5624

5625 page-ranges (1setOf rangeOfInteger(1:MAX))  
5626 IF NOT any 'rangeOfInteger' value(s) each equal to 8 octets, REJECT/RETURN 'client-error-bad-  
5627 request'.  
5628 IF first value is greater than second value in any range, the ranges are not in ascending order, or  
5629 ranges overlap, REJECT/RETURN 'client-error-bad-request'.  
5630 IF the value of the Printer object's "page-ranges-supported" attribute is 'false', copy the attribute to  
5631 the Unsupported Attributes response group and set the value to the "out-of-band" 'unsupported'  
5632 value.  
5633

5634 sides (type2 keyword)  
5635 IF NOT any single 'keyword' value less than or equal to 256-255 octets, REJECT/RETURN 'client-  
5636 error-~~bad-request~~request-value-too-long'.



5637 IF NOT in the Printer object's "sides-supported" attribute, copy the attribute and the unsupported  
5638 value to the Unsupported Attributes response group.

5639  
5640 number-up (integer(1:MAX))  
5641 IF NOT any single 'integer' value equal to 4 octets,  
5642 REJECT/RETURN 'client-error-bad-request'.  
5643 IF NOT a value or in the range of one of the values of the Printer object's "number-up-supported"  
5644 attribute, copy the attribute and value to the Unsupported Attribute response group.

5645  
5646 ~~Orientation~~orientation-requested (type2 enum)  
5647 IF NOT any single 'enum' value equal to 4 octets,  
5648 REJECT/RETURN 'client-error-bad-request'.  
5649 IF NOT in the Printer object's "orientation-requested-supported" attribute, copy the attribute and the  
5650 unsupported value to the Unsupported Attributes response group.

5651  
5652 media (type3 keyword | name)  
5653 IF NOT any single 'keyword' or 'name' value less than or equal to 256-255 octets, REJECT/RETURN  
5654 'client-error-~~bad-request~~request-value-too-long'.  
5655 IF NOT in the Printer object's "media-supported" attribute, copy the attribute and the unsupported  
5656 value to the Unsupported Attributes response group.

5657  
5658 printer-resolution (resolution)  
5659 IF NOT any single 'resolution' value equal to 9 octets,  
5660 REJECT/RETURN 'client-error-bad-request'.  
5661 IF NOT in the Printer object's "multiple-document-handling-supported" attribute, copy the attribute  
5662 and the unsupported value to the Unsupported Attributes response group.

5663  
5664 print-quality (type2 enum)  
5665 IF NOT any single 'enum' value equal to 4 octets,  
5666 REJECT/RETURN 'client-error-bad-request'.  
5667 IF NOT in the Printer object's "print-quality-supported" attribute, copy the attribute and the  
5668 unsupported value to the Unsupported Attributes response group.

5669  
5670 unknown or unsupported attribute (i.e., there is no corresponding Printer object "xxx-supported"  
5671 attribute)  
5672 IF the attribute syntax supplied by the client is supported but the length is not legal for that attribute  
5673 syntax,  
5674 REJECT/RETURN 'client-error-bad-request' or 'client-error-request-value-too-long'.  
5675 ELSE copy the attribute and value to the Unsupported Attributes response group and change the  
5676 attribute value to the "out-of-band" 'unsupported' value. Any remaining Job Template Attributes

5677 are either unknown or unsupported Job Template attributes and are validated algorithmically  
 5678 according to their attribute syntax for proper length (see below).

5679 -----

5680

5681 If the attribute syntax is supported AND the length check fails, the IPP object REJECTS the request and  
 5682 RETURNS the 'client-error-~~bad-request~~request-value-too-long' status code, else the IPP object copies  
 5683 the unsupported Job Template attribute to the Unsupported Attributes response group and changes the  
 5684 attribute value to the "out-of-band" 'unsupported' value. The following table shows the length checks for  
 5685 all attribute syntaxes. In the following table: "<=" means less than or equal, "=" means equal to:

5686	Name	Octet length check for read-write attributes
5687	-----	-----
5688	'text <del>WithoutLanguage</del> <u>WithLanguage</u> '	<= 1023 <u>AND 'naturalLanguage' &lt;= 63</u>
5689	'text <del>WithLanguage</del> <u>WithoutLanguage</u> '	<= 1023
5690	'name <del>WithoutLanguage</del> <u>WithLanguage</u> '	<= <del>255</del> <u>255</u> <u>AND 'naturalLanguage' &lt;= 63</u>
5691	'name <del>WithLanguage</del> <u>WithoutLanguage</u> '	<= 255
5692	'keyword'	<= 255
5693	<del>'keyword'   'name'</del>	<del>&lt;= 255</del>
5694	'enum'	= 4
5695	'uri'	<= 1023
5696	'uriScheme'	<= 63
5697	'charset'	<= 63
5698	'naturalLanguage'	<= 63
5699	'mimeMediaType'	<= <del>63</del> <u>255</u>
5700	'octetString'	<= 1023
5701	'boolean'	= 1
5702	'integer'	= 4
5703	'rangeOfInteger'	= 8
5704	'dateTime'	= 11
5705	'resolution'	= 9
5706	'lsetOf X'	
5707		

#### 5708 ~~15.4.4~~16.4.4 Check for conflicting Job Template attributes values

5709 Once all the Operation and Job Template attributes have been checked individually, the Printer object  
 5710 SHOULD check for any conflicting values among all the supported values supplied by the client. For  
 5711 example, a Printer object might be able to staple and to print on transparencies, however due to physical  
 5712 stapling constraints, the Printer object might not be able to staple transparencies. The IPP object copies  
 5713 the supported attributes and their conflicting attribute values to the Unsupported Attributes response  
 5714 group. The Printer object only copies over those attributes that the Printer object either ignores or  
 5715 substitutes in order to resolve the conflict, and it returns the original values which were supplied by the  
 5716 client. For example suppose the client supplies "finishings" equals 'staple' and "media" equals



5717 'transparency', but the Printer object does not support stapling transparencies. If the Printer chooses to  
5718 ignore the stapling request in order to resolve the conflict, the Printer objects returns "finishings" equal to  
5719 'staple' in the Unsupported Attributes response group. If any attributes are multi-valued, only the  
5720 conflicting values of the attributes are copied.

5721 Note: The decisions made to resolve the conflict (if there is a choice) is implementation dependent.

5722 ~~15.4.5~~16.4.5 Decide whether to REJECT the request

5723 If there were any unsupported Job Template attributes or unsupported/conflicting Job Template attribute  
5724 values and the client supplied the "ipp-attribute-fidelity" attribute with the 'true' value, the Printer object  
5725 REJECTS the request and return the status code:

5726 (1) 'client-error-conflicting-attributes' status code, if there were any conflicts between attributes  
5727 supplied by the client.

5728 (2) 'client-error-attributes-or-values-not-supported' status code, otherwise.

5729

5730 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in  
5731 this step. If the unsupported Operation attribute was a serious error, the above already rejected the  
5732 request in a previous step. If control gets to this step with unsupported Operation attributes being  
5733 returned, they are not serious errors.

5734 ~~15.4.6~~16.4.6 For the Validate-Job operation, RETURN one of the success status codes

5735 If the requested operation is the Validate-Job operation, the Printer object returns:

5736 (1) the "successful-ok" status code, if there are no unsupported or conflicting Job Template attributes  
5737 or values.

5738 (2) the "successful-ok-conflicting-attributes, if there are any conflicting Job Template attribute or  
5739 values.

5740 (3) the "successful-ok-ignored-or-substituted-attributes, if there are only unsupported Job Template  
5741 attributes or values.

5742

5743 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in  
5744 this step. If the unsupported Operation attribute was a serious error, the above already rejected the  
5745 request in a previous step. If control gets to this step with unsupported Operation attributes being  
5746 returned, they are not serious errors.

5747 ~~15.4.7~~16.4.7 Create the Job object with attributes to support

5748 If "ipp-attribute-fidelity" is set to 'false' (or it was not supplied by the client), the Printer object:

- 5749 (1) creates a Job object, assigns a unique value to the job's "job-uri" and "job-id" attributes, and  
5750 initializes all of the job's other supported Job Description attributes.
- 5751 (2) removes all unsupported attributes from the Job object.
- 5752 (3) for each unsupported value, removes either the unsupported value or substitutes the unsupported  
5753 attribute value with some supported value. If an attribute has no values after removing  
5754 unsupported values from it, the attribute is removed from the Job object (so that the normal  
5755 default behavior at job processing time will take place for that attribute).
- 5756 (4) for each conflicting value, removes either the conflicting value or substitutes the conflicting  
5757 attribute value with some other supported value. If an attribute has no values after removing  
5758 conflicting values from it, the attribute is removed from the Job object (so that the normal default  
5759 behavior at job processing time will take place for that attribute).

5760

5761 If there were no attributes or values flagged as unsupported, or the value of "ipp-attribute-fidelity" was  
5762 'false', the Printer object is able to accept the create request and create a new Job object. If the "ipp-  
5763 attribute-fidelity" attribute is set to 'true', the Job Template attributes that populate the new Job object are  
5764 necessarily all the Job Template attributes supplied in the create request. If the "ipp-attribute-fidelity"  
5765 attribute is set to 'false', the Job Template attributes that populate the new Job object are all the client  
5766 supplied Job Template attributes that are supported or that have value substitution. Thus, some of the  
5767 requested Job Template attributes may not appear in the Job object because the Printer object did not  
5768 support those attributes. The attributes that populate the Job object are persistently stored with the Job  
5769 object for that Job. A Get-Job-Attributes operation on that Job object will return only those attributes  
5770 that are persistently stored with the Job object.

5771 Note: All Job Template attributes that are persistently stored with the Job object are intended to be  
5772 "override values"; that is, they take precedence over whatever other embedded instructions might be  
5773 in the document data itself. However, it is not possible for all Printer objects to realize the semantics of  
5774 "override". End users may query the Printer's "pdl-override-supported" attribute to determine if the  
5775 Printer either attempts or does not attempt to override document data instructions with IPP attributes.

5776 There are some cases, where a Printer supports a Job Template attribute and has an associated default  
5777 value set for that attribute. In the case where a client does not supply the corresponding attribute, the  
5778 Printer does not use its default values to populate Job attributes when creating the new Job object; only  
5779 Job Template attributes actually in the create request are used to populate the Job object. The Printer's  
5780 default values are only used later at Job processing time if no other IPP attribute or instruction embedded  
5781 in the document data is present.

5782 Note: If the default values associated with Job Template attributes that the client did not supply were to  
5783 be used to populate the Job object, then these values would become "override values" rather than  
5784 defaults. If the Printer supports the 'attempted' value of the "pdl-override-supported" attribute, then these  
5785 override values could replace values specified within the document data. This is not the intent of the  
5786 default value mechanism. A default value for an attribute is used only if the create request did not specify  
5787 that attribute (or it was ignored when allowed by "ipp-attribute-fidelity" being 'false') and no value was  
5788 provided within the content of the document data.

5789 If the client does not supply a value for some Job Template attribute, and the Printer does not support  
5790 that attribute, as far as IPP is concerned, the result of processing that Job (with respect to the missing  
5791 attribute) is undefined.

5792 ~~15.4.8~~16.4.8 Return one of the success status codes

5793 Once the Job object has been created, the Printer object accepts the request and returns to the client:

- 5794 (1) the 'successful-ok' status code, if there are no unsupported or conflicting Job Template attributes  
5795 or values.  
5796 (2) the 'successful-ok-conflicting-attributes' status code, if there are any conflicting Job Template  
5797 attribute or values.  
5798 (3) the 'successful-ok-ignored-or-substituted-attributes' status code, if there are only unsupported Job  
5799 Template attributes or values.  
5800

5801 Note: Unsupported Operation attributes or values that are returned do not affect the status returned in  
5802 this step. If the unsupported Operation attribute was a serious error, the above already rejected the  
5803 request in a previous step. If control gets to this step with unsupported Operation attributes being  
5804 returned, they are not serious errors.

5805 The Printer object also returns Job status attributes that indicate the initial state of the Job ('pending',  
5806 'pending-held', 'processing', etc.), etc. See Print-Job Response, section 3.2.1.2.

5807 ~~15.4.9~~16.4.9 Accept appended Document Content

5808 The Printer object accepts the appended Document Content data and either starts it printing, or spools it  
5809 for later processing.

5810 ~~15.4.10~~16.4.10 Scheduling and Starting to Process the Job

5811 The Printer object uses its own configuration and implementation specific algorithms for scheduling the  
5812 Job in the correct processing order. Once the Printer object begins processing the Job, the Printer

5813 changes the Job's state to 'processing'. If the Printer object supports PDL override (the "pdl-override-  
5814 supported" attribute set to 'attempted'), the implementation does its best to see that IPP attributes take  
5815 precedence over embedded instructions in the document data.

#### 5816 ~~15.4.11~~16.4.11 Completing the Job

5817 The Printer object continues to process the Job until it can move the Job into the 'completed' state. If an  
5818 Cancel-Job operation is received, the implementation eventually moves the Job into the 'canceled' state.  
5819 If the system encounters errors during processing that do not allow it to progress the Job into a  
5820 completed state, the implementation halts all processing, cleans up any resources, and moves the Job into  
5821 the 'aborted' state.

#### 5822 ~~15.4.12~~16.4.12 Destroying the Job after completion

5823 Once the Job moves to the 'completed', 'aborted', or 'canceled' state, it is an implementation decision as to  
5824 when to destroy the Job object and release all associated resources. Once the Job has been destroyed, the  
5825 Printer would return either the "client-error-not-found" or "client-error-gone" status codes for operations  
5826 directed at that Job.

5827 Note: the Printer object SHOULD NOT re-use a "job-uri" or "job-id" value for a sufficiently long time  
5828 after a job has been destroyed, so that stale references kept by clients are less likely to access the wrong  
5829 (newer) job.

#### 5830 ~~15.4.13~~16.4.13 Interaction with "ipp-attribute-fidelity"

5831 Some Printer object implementations may support "ipp-attribute-fidelity" set to 'true' and "pdl-override-  
5832 supported" set to 'attempted' and yet still not be able to realize exactly what the client specifies in the  
5833 create request. This is due to legacy decisions and assumptions that have been made about the role of job  
5834 instructions embedded within the document data and external job instructions that accompany the  
5835 document data and how to handle conflicts between such instructions. The inability to be 100% precise  
5836 about how a given implementation will behave is also compounded by the fact that the two special  
5837 attributes, "ipp-attribute-fidelity" and "pdl-override-supported", apply to the whole job rather than  
5838 specific values for each attribute. For example, some implementations may be able to override almost all  
5839 Job Template attributes except for "number-up".

#### 5840 ~~15.5~~16.5 Using Job Template Attributes During Document Processing.

5841 The Printer object uses some of the Job object's Job Template attributes during the processing of the  
5842 document data associated with that job. These include, but are not limited to, "orientation", "number-

5843 up", "sides", "media", and "copies". The processing of each document in a Job Object SHALL follow the  
5844 steps below. These steps are intended only to identify when and how attributes are to be used in  
5845 processing document data and any alternative steps that accomplishes the same effect can be used to  
5846 implement this specification.

- 5847 1. Using the client supplied "document-format" attribute or some form of document format detection  
5848 algorithm (if the value of "document-format" is not specific enough), determine whether or not  
5849 the document data has already been formatted for printing. If the document data has been  
5850 formatted, then go to step 2. Otherwise, the document data SHALL be formatted. The formatting  
5851 detection algorithm is implementation defined and is not specified by this specification. The  
5852 formatting of the document data uses the "orientation-requested" attribute to determine how the  
5853 formatted print data should be placed on a print-stream page, see section 4.2.10 for the details.  
5854
- 5855 2. The document data is in the form of a print-stream in a known media type. The "page-ranges"  
5856 attribute is used to select, as specified in section 4.2.7, a sub-sequence of the pages in the print-  
5857 stream that are to be processed and images.  
5858
- 5859 3. The input to this step is a sequence of print-stream pages. This step is controlled by the "number-  
5860 up" attribute. If the value of "number-up" is N, then during the processing of the print-stream  
5861 pages, each N print-stream pages are positioned, as specified in section 4.2.9, to create a single  
5862 impression. If a given document does not have N more print-stream pages, then the completion of  
5863 the impression is controlled by the "multiple-document-handling" attribute as described in section  
5864 4.2.4; when the value of this attribute is 'single-document', the print-stream pages of document  
5865 data from subsequent documents is used to complete the impression.  
5866

5867 The size(scaling), position(translation) and rotation of the print-stream pages on the impression is  
5868 implementation defined. Note that during this process the print-stream pages may be rendered to  
5869 a form suitable for placing on the impression; this rendering is controlled by the values of the  
5870 "printer-resolution" and "print-quality" attributes as described in sections 4.2.12 and 4.2.13. In the  
5871 case N=1, the impression is nearly the same as the print-stream page; the differences would only  
5872 be in the size, position and rotation of the print-stream page and/or any decoration, such as a  
5873 frame to the page, that is added by the implementation.  
5874

- 5875 4. The collection of impressions is placed, in sequence, onto sides of the media sheets. This placement  
5876 is controlled by the "sides" attribute and the orientation of the print-stream page, as described in  
5877 section 4.2.8. The orientation of the print-stream pages affects the orientation of the impression;  
5878 for example, if "number-up" equals 2, then, typically, two portrait print-stream pages become one  
5879 landscape impression. Note that the placement of impressions onto media sheets is also controlled  
5880 by the "multiple-document-handling" attribute as described in section 4.2.4.  
5881

- 5882 5. The "copies" and "multiple-document-handling" attributes are used to determine how many copies  
5883 of each media instance are created and in what order. See sections 4.2.5 and 4.2.4 for the details.  
5884
- 5885 6. When the correct number of copies are created, the media instances are finished according to the  
5886 values of the "finishings" attribute as described in 4.2.6. Note that sometimes finishing operations  
5887 may require manual intervention to perform the finishing operations on the copies, especially  
5888 uncollated copies. This specification allows any or all of the processing steps to be performed  
5889 automatically or manually at the discretion of the Printer object.

5890 ~~46.17.~~ APPENDIX E: Generic Directory Schema

5891 This section defines a generic schema for an entry in a directory service. A directory service is a means  
5892 by which service users can locate service providers. In IPP environments, this means that IPP Printers  
5893 can be registered (either automatically or with the help of an administrator) as entries of type printer in  
5894 the directory using an implementation specific mechanism such as entry attributes, entry type fields,  
5895 specific branches, etc. IPP clients can search or browse for entries of type printer. Clients use the  
5896 directory service to find entries based on naming, organizational contexts, or filtered searches on attribute  
5897 values of entries. For example, a client can find all printers in the "Local Department" context.  
5898 Authentication and authorization are also often part of a directory service so that an administrator can  
5899 place limits on end users so that they are only allowed to find entries to which they have certain access  
5900 rights. IPP itself does not require any specific directory service protocol or provider.

5901 Note: Some directory implementations allow for the notion of "aliasing". That is, one directory entry  
5902 object can appear as multiple directory entry object with different names for each object. In each case,  
5903 each alias refers to the same directory entry object which refers to a single IPP Printer object.

5904 The generic schema is a subset of IPP Printer Job Template and Printer Description attributes (sections  
5905 4.2 and 4.4). These attributes are identified as either MANDATORY or OPTIONAL for the directory  
5906 entry itself. This conformance labeling is NOT the same conformance labeling applied to the attributes of  
5907 IPP Printers objects. MANDATORY attributes MUST be associated with each directory entry.  
5908 OPTIONAL attributes SHOULD be associated with the directory entry (if known or supported). In  
5909 addition, all directory entry attributes SHOULD reflect the current attribute values for the corresponding  
5910 Printer object.

5911 In order to bridge between the directory service and the IPP Printer object, one of the MANDATORY  
5912 directory entry attributes is the Printer object's "printer-uri-supported" attribute. The IPP client queries  
5913 the "printer-uri-supported" attribute in the directory entry and then addresses the IPP Printer object using  
5914 one of its URIs. The "uri-security-supported" attribute identifies the protocol (if any) used to secure a  
5915 channel.

5916 The following attributes define the generic schema for directory entries of type PRINTER:

5917	printer-uri-supported	MANDATORY	Section 4.4.1
5918	uri-security-supported	MANDATORY	Section 4.4.2
5919	printer-name	MANDATORY	Section 4.4.3
5920	printer-location	OPTIONAL	Section 4.4.4
5921	printer-info	OPTIONAL	Section 4.4.5
5922	printer-more-info	OPTIONAL	Section 4.4.6
5923	printer-make-and-model	OPTIONAL	Section 4.4.8
5924	charset-supported	MANDATORY	Section 4.4.15
5925	generated-natural-language-supported	MANDATORY	Section 4.4.17
5926	document-format-supported	OPTIONAL	Section 4.4.19
5927	color-supported	OPTIONAL	Section 4.4.23
5928	finishings-supported	OPTIONAL	Section 4.2.6
5929	number-up-supported	OPTIONAL	Section 4.2.7
5930	sides-supported	OPTIONAL	Section 4.2.8
5931	media-supported	OPTIONAL	Section 4.2.11
5932	printer-resolution-supported	OPTIONAL	Section 4.2.12
5933	print-quality-supported	OPTIONAL	Section 4.2.13
5934			